AD-A040 804 MODELS OF THE US ARMY WORLDWIDE LOGISTIC SYSTEM (MAWLOGS) VOLUME III MODU..(U) BDM CORP VIENNA VA FEB 77 BDM/W-76-211-TR-VOL-3-PT-6 DAAG39-76-C-0134

1/4

UNCLASSIFIED F/G 15/5 NL

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA040804

THE BDM CORPORATION

(1)

DDC
RECEIVED
JUN 22 1977

6

MODELS OF THE US ARMY
WORLDWIDE LOGISTIC SYSTEM
(MAWLOGS).
VOLUME III-MODULE CATALOG.
PART 6 - CONTINUOUS SERVICE MODULES.

AD No. ____
DDC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

THE BDM CORPORATION

1920 Aline Avenue
Vienna, Virginia 22180
Phone (703) 821-5000

(1)

(12) 276p.

(11) February 1977

(14) BDM/W-76-211-TR-Vol-3-Pt-6

(15) DAAG39-76-C-0134

(6)

MODELS OF THE US ARMY
WORLDWIDE LOGISTIC SYSTEM
(MAWLOGS).
VOLUME III-MODULE CATALOG.
PART 6 - CONTINUOUS SERVICE MODULES.

391 962

FOREWORD

This Part of the MAWLOGS Module Catalog is submitted to the Department of the Army, Washington, D.C. 20310 by the BDM Corporation, 1920 Aline Avenue, Vienna, Virginia 22180, as required by Contract Number DAAG39-76-C-0134.

This document is one of sixteen that describes the Models of the US Army Worldwide Logistic System (MAWLOGS). MAWLOGS was developed for the Deputy Chief of Staff for Logistics, Department of the Army, under the monitorship of the US Army Logistics Evaluation Agency and the US Army Logistics Center. The development objective was to provide a capability to analyze and compare the performance of multifunctional logistic systems, to include both current and proposed systems. MAWLOGS is not a model of a particular Army logistic system. It is a system for the rapid assembly of discrete-event stochastic simulation models of a wide range of logistic systems and for the processing and interpretation of data associated with the execution of such models. The original documentation was completed in 1974. Documentation for subsequent software development has added five volumes to the original eleven. The documents describing the system and how to apply it are listed below.

- Volume I - General Description AD 920 692
- Volume II - User's Manual
- Volume IIA - Addendum to User's Manual B017422
- Volume III - Module Catalog 1/923112L
  - Part 1 - Service Modules 923116?
  - Part 2 - Field Maintenance and Supply Modules AD923117L 2/923124L
  - Part 3 - Wholesale Supply and Maintenance Modules AD 923118
  - Part 4 - Transportation Modules AD923131L
  - Part 5 - Communication Modules AD923125L
  - Part 6 - Continuous Service Modules A040804
  - Part 7 - Continuous Supply Modules A040805
  - Part 8 - Containerization Modules A040806
  - Part 9 - Model Change Modules

ii blank

Volume IV - Programmer's Guide

Part 1 - Writing and Testing Modules, Module Library Maintenance and General Guidance AD923126L

Part 2 - Technical Description of the Model Assembler Program AD923127L

Part 3 - Technical Description of the Output Data Postprocessor System and Programs AD923132

Part 4 - Module Library Program Listings

Volume IVA - Addendum to Programmer's Guide AD A040807

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

x - blank

LIST OF FIGURES

xii blank

LIST OF TABLES

CHAPTER I
INTRODUCTION TO MODULE CATALOG

The MAWLOGS System is a set of computer programs by which a large number of stochastic, discrete-event, computerized simulation models of logistic systems or portions of systems can be generated. The computer programs in the MAWLOGS System may be divided into four groups, each of which constitutes a basic element of the System. One, called the Module Library, contains programs that simulate logistic activities or perform simulation bookkeeping activities. A second group of programs, called the Model Assembler, generates MAWLOGS models by retrieving required modules from the Module Library and creating sufficient additional programs to link the selected modules together into a model. The other two groups constitute the Output Data Postprocessor System and the Automated Input Data System. An overview of the MAWLOGS System is given in Volume I and, in briefer form in Section I of Volume II.

The MAWLOGS Module Catalog is the directory to the Module Library. Its purpose is to describe to the user of the MAWLOGS System the modules that are in the library and the logistic or simulation function each performs. For logistic modules, such additional information as the level of detail, decision rules used, and statistics collected are given.

The User's Manual, Volume II of the four-volume MAWLOGS documentation, describes how to define a model to the Model Assembler in terms of modules, how to run the Assembler to produce the model, how to use the model, to include preparation of inputs for simulation control, and how to use the MAWLOGS Output Data Postprocessor to analyze model outputs. Volume IV, "Programmer's Guide," contains technical descriptions of the Model Assembler and the Output Data Postprocessor System. The Programmer's Guide also discussed the writing of modules.

Thus, the user of the Module Catalog is likely to be working from either of two directions. The more usual case will be that of the model writer using the User's Manuals as a guide and the Module Catalog as a

reference from which to select modules appropriate for use in his model. The other case is that of the module writer who wishes to write new modules or modify existing ones. The module writer will be using the Programmer's Guide as his guide, and the technical details in the Module Catalog to determine what to modify in existing modules or what specific points must be accommodated to make any new modules compatible with existing modules.

A. CATALOG LAYOUT

The MAWLOGS Module Catalog is organized into Parts, each of which, with the exception of Part 1, describes the modules for a logistic functional area. Part 1 describes the simulation service modules. In the initial catalog, Part 2 covers field maintenance and supply, Part 3 covers wholesale maintenance and supply, Part 4 covers transporation, and Part 5 covers communications. A Part is organized into four major sections in addition to this introduction. Sections 4 and 5 describe the modules. Sections 2 and 3 present information common to large groups of modules under the headings MODULE FAMILIES and DATA STRUCTURE.

1. Module Descriptions

It is essential to distinguish between two types of modules -- those that may be referenced in a model description, called verbs, and those that may not, called routines. The catalog descriptions of verbs for a logistic functional area will be found in the fourth section of a Part, entitled "VERBS," the descriptions of supporting routines in the fifth, entitled "ROUTINES." Within each section the modules are in alphabetical order. Each Part contains an index to its verbs and supporting routines.

A module of either type is documented in a standard format, using a certain amount of phraseological and notational convention. The format and conventions are described further in subsequent paragraphs.

2. Module Families

The presence of a section called "MODULE FAMILIES" reflects recognition of the fact that modules are designed in groups along certain lines

chosen by the designer. Thus, the designer chooses the scope and level of detail with which he will cover a set of logistic activities, to what degree he will "modularize" the coverage, which options he will provide, the data structures he will use, and a variety of other details in the programming approach. He will base his work on one or more major assumptions and objectives. A knowledge of the background and viewpoint underlying a group of related modules can greatly facilitate their proper use. The MODULE FAMILIES section is intended to provide this background and overview. Examples of nodes or subnodes in which the verbs of a family are used in their intended patterns should also be sought in the MODULE FAMILIES sections. The types of statistics collected in a module family are tabulated all together in this section.

3. Data Structure

The section called "DATA STRUCTURES" contains descriptions of the organization of the data storage areas that support a module family. The description consists of the names of common blocks and, for each, the names and definitions of variables. Highlights and key points of the data structure are described. Normally there will be a major section to describe permanent attributes and a lesser one to describe temporary attributes.

B. CATALOG CONVENTIONS

A verb description in the catalog follows the outline shown below.

		VERBNAME
VERBNAME (arguments)	simple nonsimple	/function/family
		month/year written

General Description
Assembler Inputs
Examples
Statistics Collected
GASP Files Used
Permanent Attributes Accessed
Verb Inputs
Verb Outputs
Programs Called
Input/Output Files Used

Supporting routines are described in a similar format with the inapplicable headings "Parameter Slots" and "Examples" omitted. Note that a listing of the computer program is not included; these are available in Part 4 of Volume IV, Programmer's Guide. Further discussion of the content of each section of the outline, to include particular notational or descriptive conventions used, are given next. The sections progress from general to detailed. For most verbs, the model writer should not have to go beyond the section called "Statistics Collected."

1. <u>Upper Righthand Corner</u>

Here a coded summary of the verb is given in two lines under its name. The first line identifies the verb as simple, S, or nonsimple, N. It also gives a code for the logistic or other function represented and, following this, a code for the module family it is considered to be a part of. Values for the function and family codes are given in the accompanying tabulation. The second line under the verb name gives the month and year when written.

Functional areas		Module families	
Code	Area	Code	Family
C	communications	C1	communications
LS	logistic service	CH	change logic
M	maintenance	CS	continuous service
R	rebuild	LS1	logistic service
S	supply	M1	field maintenance
SS	simulation service	M2	wholesale maintenance
ST	statistics collection	PDS	permanent datasets
T	transporation	R1	field rebuild
U	user	S1	field supply
V	salvage	S2	wholesale supply
X	direct exchange	S3	revised field supply
		S4	continuous supply
		SS1	simulation service
		ST1	statistics collection
		T1	aggregate transportation
		T2	detailed transportation
		T3	containerization
		U1	fleet user
		V1	salvage
		X1	field direct exchange
		X2	revised DX

2. General Description

This section contains a general discussion of what the verb does, to include any significant mathematics. The flow of control when execution of the verb is completed is also described. The phrases "to the calling program" and "to the time file" are used for the flow of control when, respectively, a CALL RETLOG returns control to the previous stage of the logic flow that led to the verb, or a CALL ZFADE interrupts the present flow and returns control to program MAWGSP, which then removes the next event from the time file. For modules which read cards, write reports, or read or write other external files, the card formats, report formats, or file formats are described.

3. Assembler Inputs

Here the information to be specified for the verb when it is referenced in a model description written for input to the Model Assembler is defined. Two types of information are identified, arguments and parameter slots. Arguments are variables for which values are specified in the form "P = i, j, ..." i.e., they are variables whose names will appear as arguments of the verb in its FORTRAN subroutine form. Parameter slots, abbreviated "PS," are places in the logic of a verb where access to node references and the logic of other verbs may be provided for by the Model Assembler. The content of a parameter slot is described in general terms. It will be recalled from the User's Manual that a PS need not be filled. If it is filled, it is incumbent on the user to insure that the verbs chosen as content of the PS are compatible with each other and with the verb in whose PS they appear. In particular, this means that the verbs chosen for the PS must not only represent the general functional content required, but that their inputs as described in the section of this outline called Verb Inputs must be compatible with the outputs of the using verb described in the section of this outline called Verb Outputs and vice versa.

Some parameter slots are for sending something out -- a message or a shipment, for example. Such a parameter slot will often be marked "D-delay" or "R-delay." These designations represent cases in which, respectively, (a) control is expected to be given to the time file after

the arrival event has been scheduled, and (b) control is to be returned to the verb containing the PS after the arrival event has been scheduled. The prototypical verbs of these types are DELAY and RTURN. The corresponding communications verbs are COMMD and COMMR, and the corresponding transportation verbs are TRAND and TRANR. An important point to note is that in the case of a PS marked "R-delay," the presence of a RTURN or similar verb is mandatory, even if the delay time is to be zero, while in a "D-delay" PS, the DELAY or similar verb may be omitted if the delay is to be zero or if some other disposition of the entity normally thought of as being "sent" is to be provided for.

4. Examples

Examples of how the verb is used in portions of model descriptions are shown or described. Certain limitations or options are often highlighted under this heading.

5. Statistics Collected

The variables on which statistics are collected are listed in terms of a descriptive phrase and, for variables observed through calls to STATI, the names of the variables and common blocks in which the statistics indicator codes and indexes are stored. However, the statistics type code, resource identifier, and node code by which a statistics variable is identified are not given here; they must be sought in the Module Family section of the Part of the catalog in which the verb is described.

6. GASP Files Used

The GASP files accessed by the module are identified, primarily in terms of a general description of the content of the file and the names of variables and common areas in which the GASP file numbers are stored. The indirect identification of the file number reflects the fact that in general the file numbers are assigned to a particular queue when the first entry is to be filed or when the logistic functional area is initialized. Thus, the file number assignments will vary among models and runs of the same model. When a file number is stored in PERMAT in an attribute set of type QUEUE(fcntyp), the form "QUEUE(fcntyp).n" is used to identify the location where a file number is stored. In this notation n is the index

of the element in the attribute set that contains the GASP file number and "fcntyp" is a function type code such as MNOD or SNOD. The attributes of a file entry and the ranking scheme for a file are defined under the module description headings Verb Inputs and Verb Outputs.

7. Permanent Attributes Accessed

This and the next two headings -- Verb Inputs and Verb Outputs -- are related in that they all deal with data transmission among modules and between modules and the data storage areas. Thus, an initial point is that they all deal with internal data transmission as distinguished from card inputs or other input or output involving external files.

In discussing this internal data flow, it is useful to distinguish between permanent and temporary data. The phrase "permanent data" refers to data stored in locations whose variable names and the system characteristics they represent are fixed throughout a run. All of the data in PERMAT, in other permanent attribute storage areas, and in the statistics counters are of this type. The phrases "temporary data" refers to all data that are not permanent. It includes such things as the content of array POOL through which PERMAT attribute sets are transmitted, the content of arrays ATRIB and HOLD through which attributes of temporary entities stored in the GASP filing array are transmitted, and the variables IQ, KRA, and KRB, used to designate GASP file numbers and ranking schemes. Utility variables IZSWT and ZSWT in /ZMAWSY/ are also often used for temporary data. The dynamic storage areas in which the GASP files and MAWLOGS stacks are stored also contain temporary data, but these areas are accessed directly only by a few simulation service routines and so are not often referred to in the module descriptions.

The term "permanent attributes" refers to those elements of permanent data that characterize aspects of the logistic system being simulated. Examples are policy parameters such as a reorder point, resource characteristics such as the price of an item, and the capacity and response characteristics of such system elements as transporation links or materiel handling

and storage points. Data such as statistics counters and statistics indicators and indexes, or the GASP file descriptors stored in /GSPFIL/ are examples of permanent data that are not permanent attributes.

Two other terms remain to be defined: "module inputs" and "module outputs." A module input is any variable whose value (a) was determined outside the module and (b) influences the logical path taken in the module or the amount by which a module output variable will be changed. A module output is any variable whose value may be changed by the module. Inputs and outputs may be either permanent or temporary data. Further discussion of inputs and outputs may be found in the next two subsections.

Here, it will suffice to summarize the overall conventions governing the content of the three sections: Permanent Attributes Accessed, Verb Inputs, and Verb Outputs. Permanent Attributes Accessed lists all permanent attributes retrieved by, stored by, or modified by a module. Verb Inputs does not list all inputs; it lists only inputs that are not listed under the Permanent Attributes Accessed and Statistics Collected headings. Verb Outputs lists all outputs, temporary and permanent, including any entries under Permanent Attributes Accessed whose values may be changed by the module. The objective here is to minimize the multiple mention of a variable while retaining the utility of the three headings under discussion.

The purpose of the present heading, Permanent Attributes Accessed, is to list in one place those principle variables, by which the status of the logistic system is represented in the model, that have interplay in a module. From the point of view of the student of the logistic system, it is just this aspect of the module that is of interest. An element of an attribute set in PERMAT is referenced in the following notation -- FUNC(arg).m, where FUNC is a PERMAT function name representing an attribute set type, such as SICOM, arg is the argument type, such as "item," and m is the index of the element in the attribute set.

8. Verb Inputs

The data described under this heading are a subset of the total internal verb inputs. Categories not described are the statistics indicators and indexes implied by the entries under the heading Statistics Collected

and any inputs under Permanent Attributes Accessed. Inputs that are listed include primarily temporary data. A major type of temporary input is the set of attributes associated with a temporary entity, usually received in arrays ATRIB or HOLD. Inputs are listed under major subheadings indicating their source, such as Calling Program, PS1 (parameter slot 1), Verb DISTR. The purpose of the Verb Inputs section is to facilitate determination of whether the verb will fit in a particular parameter slot and to facilitate the writing of new verbs which may be used in conjunction with this verb.

9. Verb Outputs

The general purpose of this section is similar to that for Verb Inputs -- to facilitate determining whether one verb meshes with another and to facilitate the writing of new verbs that mesh with existing verbs. The outputs are listed under major subheadngs representing their destination -- PS1, PS2, ..., Calling Program, Permanent Attributes, and any other suitable headings such as the names of subroutines or verbs called directly. Only variables the module was designed to affect are listed. Thus, for example, if a set of temporary attributes is input to the module and is not changed by it, they are available for use by a subsequent module and in this sense may be thought of as having been "output" by the current module. But if the current module will never change any of the attributes and is not naturally thought of as "producing" an event of the type represented by the input attributes, these attributes will not be listed under Verb Outputs. The same is true of any other temporary data that are input but not changed.

10. Programs Called

Any programs called directly by the module are named here, with the exception of a few whose use is very common. These are CALLOG, RETLOG, and LINK for stack accessing; STAT1 for statistics collection; and FPOOL, GPOOL, PPOOL, CHPMT, PRMT, and SETPMT for PERMAT accessing.

11. Input/Output Files Used

The files referred to here are external files as distinguished from GASP files. They are identified in terms of their FORTRAN logical file numbers and general contents.

CHAPTER II
FAMILY DESCRIPTION

A. OVERVIEW

1. System Dynamics

The service modules described in this section of the catalog provide a means for employing the system dynamics simulation technique in MAWLOGS models. The system dynamics viewpoint was developed by Forrester in the late 1950s. Its principal exposition remains the original book by Forrester, Industrial Dynamics[1], published in 1961. Subsequent applications to the modeling of a city[2] and the world[3,4] led to a broadening of the name from industrial dynamics to system dynamics.

System dynamics views a system as a collection of interacting rates and levels. Material, money, and information flow into and out of accumulations of them. The levels of the accumulations affect the rates of flow, and the rates of flow affect the levels. Flows from one accumulation to another are typically subject to delay. In a real system the typical relations through which the rates and levels interact are operating policies and procedures and processing and storage capacities.

In a system dynamics model time is advanced in even steps. At the end of each time step, levels are updated by determining the net change resulting from multiplying the current values of the rates that affect them by the length of the time step. After all levels are adjusted, new rates are computed and are assumed to apply for the next time step. Thus, the flows are represented as being continuous. Their rates and related levels

---

[1] Forrester, Jay W., Industrial Dynamics, Cambridge, Mass., The M.I.T. Press, 1961.

[2] Forrester, J.W., Urban Dynamics, Cambridge, Mass., The M.I.T. Press, 1969.

[3] Forrester, J.W., World Dynamics, Cambridge, Mass., Wright-Allen Press, Inc., 1972.

[4] Meadows, D.H., D.L. Meadows, J. Randers, and W.W. Behrens, III, The Limits to Growth, Washington, D.C., Potomac Associates, 1972.

are also considered to vary continuously. But, for computational convenience, the rates are approximated by a sequence of constant values occurring over suitably short time steps, and the levels are subjected to discrete adjustments at the end of each time step.

2. Adaptation to MAWLOGS

The motivation for incorporating a continuous simulation capability into MAWLOGS was the need for a reasonably natural facility for representing highly aggregated resources and transaction streams. System dynamics was adopted as the particular methodology because it was developed with that orientation and because of the simplicity of its terms of reference - the rates, levels, and delays.

The system dynamics methodology has been implemented in the DYNAMO[5] simulation language. But that implementation could not be used directly in MAWLOGS for two reasons. First, MAWLOGS is currently designed to produce FORTRAN models. Second, and of greater importance, MAWLOGS represents a system as a network of functional entities called nodes, while DYNAMO and the expositions by Forrester lack this dimension. In their model view a system is a single, integrated, aggregated entity without visible geographic or organizational dispersion.

The adaptation of system dynamics to MAWLOGS required the programming of the delay modules in FORTRAN, the development of a means for distributing system dynamics representations over a node network and associating flows between nodes with connecting links, and a facility for collecting statistics on rates and levels as flexible as that for the discrete event logic in MAWLOGS and compatible with the MAWLOGS Postprocessors. An additional requirement was the development of a time stepping scheme that could be integrated with the discrete-event time control logic so that models containing both discrete event and system dynamics elements could be constructed.

---

[5]Pugh, Alexander L., III, DYNAMO User's Manual, Cambridge, Mass., The M.I.T. Press, 1963.

With the exception of some undeveloped statistics collection options, the system dynamics simulation methodology has been incorporated in MAWLOGS. The implementation is fully compatible with the MAWLOGS network view of a system, the (FORTRAN) modular view of the MAWLOGS Model Assembler and model description language, and the discrete event time control logic. The statistics collection capabilities provided are highly flexible and fully compatible with the MAWLOGS Postprocessors. Input, reports, restart, and model trace modules compatible with MAWLOGS procedures for these activities are also included. The service routines and data structures in which these capabilities are provided can be used as the basis for any functional module family expressed in system dynamics logic. To date a family of supply modules (see Part 7 of the MAWLOGS Module Catalog) and portions of a family of containerization modules (see Part 8) have been developed with system dynamics logic.

3. Subfamilies

It is convenient to organize the subsequent discussion around six principal subfamilies of system dynamics service modules. The subfamilies are identified in Table II-1. They encompass all 78 modules of the MAWLOGS system dynamics service module family.

As an aid to interpreting module names in this family, it is noted that the phrase "rate/level," abbreviated RL, is frequently used as a synonym for system dynamics. Less frequently used synonyms include "flow," "continuous flow," and "continuous".

B. SIMULATION CONTROL

The modules developed to control the definition of system dynamics nodes and links, the advance of simulation time, and the updating of system dynamics variables representing the status of the system being simulated are listed in Table II-2. A view of the place of system dynamics control within the overall MAWLOGS control scheme is given in Figure II-1.

TABLE II-1. PRINCIPAL SUBFAMILIES

NAME	MODULES	DESCRIPTION
SIMULATION CONTROL	9	SCHEDULING SYSTEM DEFINITION, CHANGES TO LINKAGE STRUCTURE, AND RATE/LEVEL UPDATES
INITIALIZATION AND RESTART	9	INITIALIZATION OF VARIABLES VIA BLOCK DATA PROGRAM AND INPUT MODULES; ALSO SAVE STATUS AND RESTART MODULES
LINKAGE	26	DEFINITION, MODIFICATION, AND ACCESSING OF DATA REPRESENTING FLOW LINKS BETWEEN SYSTEM DYNAMICS NODES OR SUBNODES
RATES	5	ACCESSING AND UPDATING FLOW RATES ON SYSTEM DYNAMICS LINKS
DELAYS	18	DEFINITION, MODIFICATION, AND ACCESSING OF SYSTEM DYNAMICS FLOW DELAYS
STATISTICS	11	DEFINITION, MODIFICATION, AND COLLECTION OF STATISTICS ON SYSTEM DYNAMICS RATES AND LEVELS

TABLE II-2. SIMULATION CONTROL MODULES

MODELS	DESCRIPTION
	VERBS
vMLOOP	LOOPS THROUGH FLOW CLASSES IN DATASET MGTCL(RLNOD), PERMITTING ACTION ON EACH IN TURN THROUGH PS1
vPLOOP	LOOPS THROUGH PRIORITIES IN DATASET PRIORY(RLNOD), PERMITTING ACTION UNDER EACH IN TURN VIA PS1
vRLCTR	PRINCIPAL CONTROL MODULE FOR SYSTEM DYNAMICS LOGIC - SCHEDULES UPDATES OF LEVELS AND RATES FOR EACH TIME STEP, AND SCHEDULES EXECUTIONS OF SETUP LOGIC WHEN NODES AND FLOW LINKS ARE TO BE DEFINED OR REVISED
vRLMSU	SETS SWITCH RLMODE TO SETUPM TO SIGNAL SETUP STEP
vRLNOD(RLNAM,RLN)	SETS RLNOD AND KRLNOD IN /RLSYS/ TO RLN, THE IDENTIFYING NUMBER OF THE CURRENT SYSTEM DYNAMICS NODE; DURING SETUP PASS STORES RLNAM IN RLNNAM(RLN) IN /RLSYS/; PRINTS TRACE INFORMATION WHEN RTRACE IN /ZMAWSY/ IS ON
vRLOOP(RTDST)	USING LRLOOP, LOOPS THROUGH ALL DATASETS OF TYPE RTDST AT RLNOD, AND EXECUTES PS1 FOR EACH IN TURN
vRLSU	USED IN PS2 OF vRLCTR FOR ITS MNEMONIC VALUE OF INDICATING INITIATION OF A SETUP PASS THROUGH THE SYSTEM DYNAMICS PORTIONS OF A MAWLOGS MODEL
vTSTEP	DETERMINES SIMULATION TIME AT WHICH NEXT UPDATE OF LEVELS AND RATES IS TO OCCUR AND SCHEDULES INITIATION OF THAT UPDATE THROUGH THE MAWLOGS TIME FILE
	ROUTINES
LRLOOP(RDSTYP,RATPS)	LOOPS THROUGH LINKED-RATE DATASETS OF TYPE RDSTYP AT NODE RLNOD, RETURNING THE ELEMENTS OF EACH IN TURN IN ARRAY RATPS

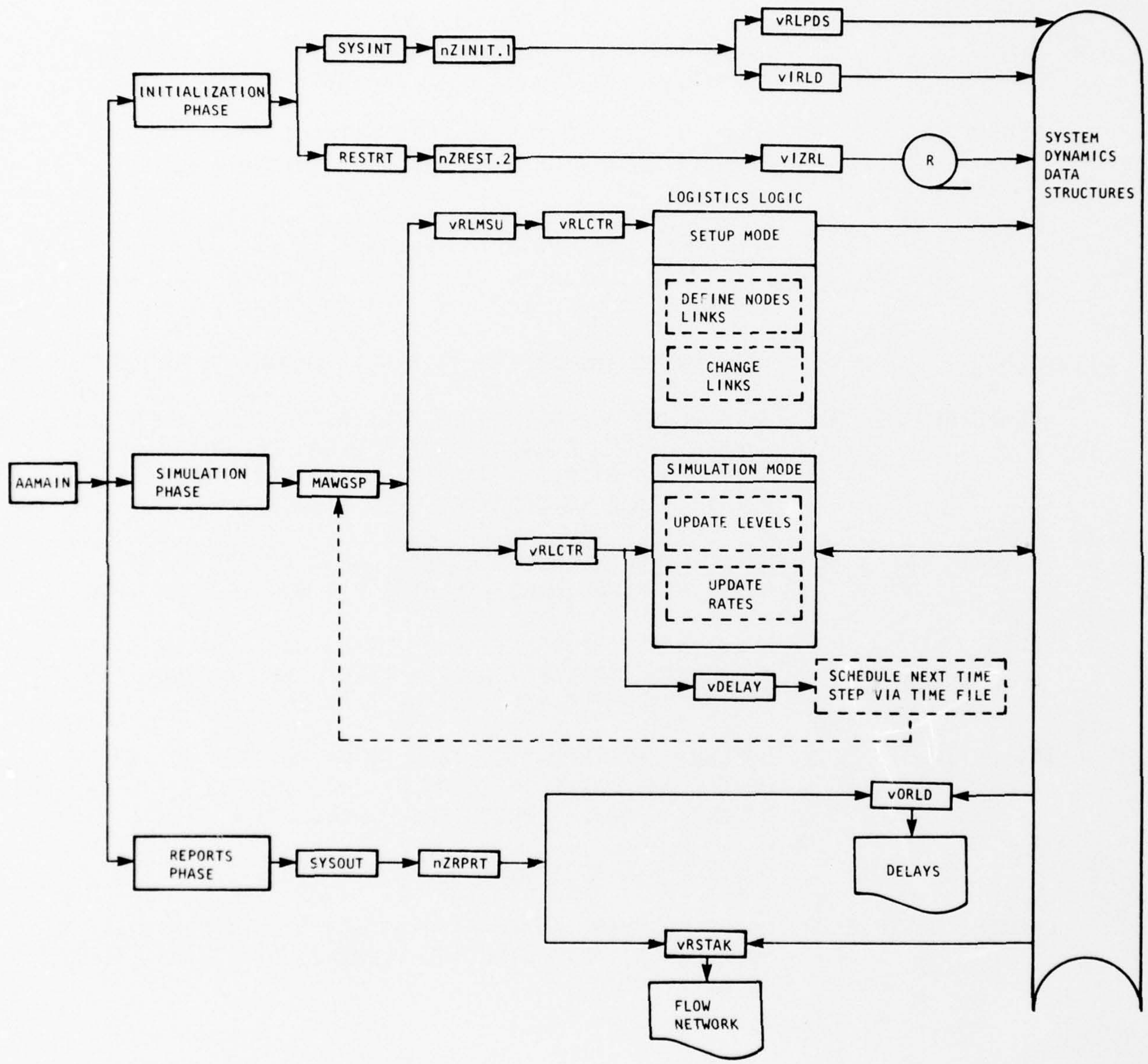


Figure II-1. System Dynamics Logic in MAWLOGS

1. <u>Time Stepping</u>

Levels and rates are updated at the end of each time step. In a MAWLOGS model containing both discrete event and system dynamics elements, a variety of other, discrete event status variables will be updated at randomly occurring times in the interval spanned by one system dynamics time step. To insure proper sequencing of the updating of discrete event and continuous variables, the update events for continuous variables are scheduled via the MAWLOGS time file, the same mechanism as is used to schedule other events. This is done by verb TSTEP via PS1.

2. <u>Updating Rates and Levels</u>

Assume that rate R represents an input stream for level L. Let TSTEP be the time step in units of simulation time. Then at the end of each time step, the current rate R is multiplied by TSTEP to determine the amount by which L should be incremented to reflect the influence of rate R during the time step just ended. After all levels have been updated in this way, all rates that are affected by levels are set to reflect the updated levels. The new rates are assumed to apply during the next time step and are used in the next adjustment of levels, to be performed at the end of the next time step.

In a MAWLOGS model containing system dynamics logic, rates and levels equations will generally be scattered among a number of nodes and subnodes. More precisely, they will be scattered among a number of modules, each of which may be invoked in a number of nodes. To insure that the equations are executed in proper order, the following procedures were adopted:

(1) Every system dynamics verb must contain both a levels section and a rates section (either of which may be empty), each terminated by a CALL RETLOG; a branching instruction that uses RLSW in /RLSYS/ to determine which section is to be executed in a particular call must also be included

(2) Every node or subnode containing system dynamics logic must transfer control to another system dynamics node or subnode or the rate/level control node after each execution of its own levels or rates equations

(3) The model must contain a rate/level control node (any node name may be used) in which verbs RLCTR and TSTEP are used to initiate the levels and rates update cycles through the nodes and to schedule them at the end of each time step.

To illustrate these ideas, consider a model containing two logistics or other functional nodes in addition to the system dynamics control node. The model description is shown in Figure II-2. Note that a node cycle SDYNC → A → B → B.2 → SDYNC → ... is defined by the node references. With the proper logic in RLCTR, it is now possible to cause execution of this cycle twice for each time step - once for updating levels, then once for updating rates. Note that subnode A.2 is not in the cycle. It is assumed that modules I, J, and K do not contain system dynamics logic, hence they should not be included in the system dynamics update cycle.

```
+ NODE A
     A. X, Y, *B
     /2/ I, J, K              $
+ NODE B
     B. X, Y, *B.2
     /2/ Z, *SDYNC            $
+ SYSTEM DYNAMICS CONTROL NODE
     SDYNC.RLCTR (1=TSTEP(1=DELAY(P=0),*SDYNC)), *A  $
```

Figure II-2. Node Linking for Rate/Level Updating

The modules X, Y, and Z must be organized in accordance with convention (1) noted above. This organization is schematized in Figure II-3. Some modules will contain only levels or only rates equations; they may have a slightly simpler form.

```
/* X, S, O-
---------------------------------------------
      COMMON/RLSYS/..., RLSW, LEVELS, RATES,...
---------------------------------------------
      IF (RLSW.EQ.LEVELS) GO TO 20
      IF (RLSW.NE.RATES) ERROR? RETLOG?
---------------------------------------------
C  LEVELS EQUATIONS
      10 CONTINUE
           .
           .
           .
      CALL RETLOG
---------------------------------------------
C  RATES EQUATIONS
      20 CONTINUE
           .
           .
           .
      CALL RETLOG
---------------------------------------------
      END
```

Figure II-3. Structure of System Dynamics Verb in MAWLOGS

The first execution of the system dynamics control node will normally be scheduled exogenously via verb FILE in node ZINIT, not shown in the example of Figure II-2.

3. Setup Mode

The discussion to this point has focused on model control in the course of simulating system behavior. It has described use of the MAWLOGS time file to schedule system dynamics update cycles, use of a system dynamics control node and standard MAWLOGS node-linking notation to define a cycle through all portions of the model containing system dynamics logic, and adoption of a simple standard structure for system dynamics verbs. Those techniques are considered to be part of the simulation mode of operation.

A second mode - the setup mode - is also defined for system dynamics in MAWLOGS. It provides for the definition or modification of system dynamics flow links for information or material. The interconnections among nodes are most conveniently defined piecemeal, i.e., from the local vantage point of individual nodes, as is done in discrete event MAWLOGS models. The MAWLOGS system dynamics setup mode is designed to provide that feature by causing execution of a levels pass and a rates pass through the system dynamics node cycle with the RLMODE switch set to SETUPM. In simulation mode it is set to SIM. To take advantage of the setup opportunity, a system dynamics verb must incorporate a branching instruction based on RLMODE in /RLSYS/ and a setup section of logic in addition to the RLSW branching and the rates and levels shown in Figure II-3.

Verbs RLSU and RLMSU and PS2 of verb RLCTR implement the setup option. The first node cycle in a model run started from scratch is assumed to be a setup cycle, i.e., RLMODE is initialized to SETUPM. Subsequent setup mode cycles can be triggered endogenously or scheduled exogenously by invoking vRLMSU to set RLMODE = SETUPM and transferring control to the system dynamics control node.

To illustrate, consider Figure II-4. It is an embellished version of Figure II-2. The system dynamics control node, SDYNC, has been expanded

```
+ NODE A WITH ENDOGENOUS TRIGGERING OF RL SETUP PASS
     A.  X, Y, *B
     /2/ I, J, K (1 = RLMSU, *SDYNC)  $

+ NODE B
     B.  X, Y, *B.2
     /2/ Z, *SDYNC  $

+ SYSTEM DYNAMICS CONTROL NODE
     SDYNC.  RLCTR (1 = TSTEP (1 = DELAY (P = 0), *SDYNC)  $
                    2 = RLSU (1 = LOGIC TO CONCLUDE SETUP), FADE),
          *A  $

+ INITIALIZATION NODE WITH EXOGENOUS SCHEDULING OF RL SETUP PASS
     ZINIT. . . .,
          FILE (P = i $ 1 = RTURN (P = 0), RLMSU, *SDYNC), . . .  $
```

Figure II-4. Scheduling System Dynamics Setup Passes

to include PS2 of verb RLCTR. PS2 is executed at the conclusion of a setup pass. (Examples of content for PS1 of RLSU will be given later, in the discussion of flow links.) Node A.2 has been expanded to illustrate how a module, K in this instance, might be used to trigger a setup pass when it detects conditions that cause it to execute PS1. Note that it is not itself a system dynamics module. Any module, system dynamics or not, may be used this way to trigger a setup pass. When a system dynamics module is used, it is important to allow the rates/levels update sequence to finish before initiating the setup pass. Thus, if K were actually a system dynamics module in node A.1, it would be necessary to modify the model description as follows:

A. X, Y, K(1=RTURN(P=0), RLMSU, *SDYNC), *B

The RTURN (P=0) preceding verb RLMSU causes the setup pass to be the next event after completion of the current update passes. Node ZINIT has been included to show how verb FILE can be used for exogenous scheduling of setup passes. All that is required in both the exogenous and endogenous cases is that vRLMSU be invoked to set the mode switch and then that control be transferred to the system dynamics control node.

4. Node Identification and Logic Tracing

To improve model readability and to insure ready availability of a node identifier that can be used to define flow links and associate data with nodes and links, verb RLNOD is provided for use at the beginning of each system dynamics node or subnode for which unique identification is desired. When a node or subnode is entered via vRLNOD, variables RLNOD and KRLNOD in /RLSYS/ are automatically set to the identification number specified as the second argument of vRLNOD. Of course the standard MAWLOGS node identifier, IZNODE in /ZMAWSY/, can also be used, but it cannot be freely configured nor can it distinguish among subnodes or be changed.

Module vRLNOD also checks variable RTRACE in /ZMAWSY/ and prints logic trace information for the current node when RTRACE equals the node number. Further discussion and an example of the RTRACE capability are given in the addendum to the MAWLOGS User's Manual in which use of the system dynamics (or "flow" or "rate/level") facilities of MAWLOGS is described.

5. Looping Aids

To facilitate at a node the updating of multiple sets of rates and levels of the same generic composition but differing in the particular resource or priority represented, verbs RLOOP, MLOOP, and PLOOP evolved. PLOOP steps through the priorities in PDS dataset PRIORY(RLNOD) and for each in turn executes PS1 of PLOOP. MLOOP steps through the resource categories identified in PDS dataset MGTCL(RLNOD) and for each executes PS1 of MLOOP. The resource categories in MGTCL are generally regarded as subcategories of a more highly aggregated category. Examples are splitting the total material consumed by a node into classes of supply, or splitting a class of supply into portions representing its suppliers.

Verb RLOOP represents a more elaborate looping option than PLOOP or MLOOP. It is based on the system dynamics flow links data structure in which flow paths leading into and out of a node are associated with the node by means of pointer elements in PDS datasets categorized as rate-linking datasets. The idea is that a flow rate that traverses a path of one or more links is associated (linked) with its node of origin (or destination for *return flow paths*) through these datasets. (In the continuous supply module family EXRAT and FCRAT are rate-linking dataset types.) RLOOP retrieves for a node every dataset of the type specified as an argument, and for each invokes PS1 of RLOOP. RLOOP is an invaluable aid for insuring that all rates entering a node from outside it are incorporated in the updating of levels and rates. The completeness and efficiency of RLOOP are insured by the existence of one dataset of type LDSC(RLNOD) for every node that is part of a flow path. See the discussion of linkage for further details. Routine LRLOOP is the workhorse by which vRLOOP accesses the flow link data structure.

C. INITIALIZATION AND RESTART MODULES

The system dynamics module family includes block data and input modules for initializing variables in a cold start of a model and in a restart. They are listed in Table II-3.

TABLE II-3. INITIALIZATION AND RESTART MODULES

MODULE	DESCRIPTION
	INITIALIZATION
BLKRL	BLOCK DATA ROUTINE FOR SYSTEM DYNAMICS DATA AREAS
DATAN	READS TSTEP AND NSTW AMONG THE USUAL MAWLOGS MODEL CONTROL PARAMETERS
vIBXST	INITIALIZES STATISTICS COLLECTION BOXCAR TRAINS
vIRLD	INPUT MODULE FOR SYSTEM CYNAMICS DELAYS
vIZRL	RESTART INPUT MODULE FOR SYSTEM DYNAMICS DATA STRUCTURES (READS FILE CREATED BY vSVRL)
RLINT	CONVERTS CERTAIN FLOATING POINT INPUTS READ BY vRLPDS TO INTEGER
vRLPDS	PRINCIPLE INPUT MODULE FOR SYSTEM DYNAMICS DATA - INCLUDES ASSIGNMENT OF STATISTICS COLLECTION PATTERN
RLSASN	ASSIGNS A POSITION IN ARRAYS KRLST AND KRLSTI FOR A SYSTEM DYNAMICS STATISTICAL VARIABLE, AND ASSIGNS IDENTIFIERS TO ARRAY RLSNAM, TCTNAM, CLTNAM, AND HNAM AS APPROPRIATE
cRLSIZ	"COMMON" MODULE, INVOKED IN BLKRL, IN WHICH DATA STATEMENTS SET DIMENSIONING VARIABLES TO VALUES DEFINED IN THE MODEL ASSEMBLER DIMENSIONS MODULE
RLSSET	SETS UP ARRAYS KRLST AND LRLST IN /RLSYS/ FOR USE BY RLSTAT
vSVRL	RESTART SAVE MODULE - SAVES ON A FILE CURRENT STATUS OF OF SYSTEM DYNAMICS PORTIONS OF MODEL FOR LATER USE IN RESTART (TO BE READ BY vIZRL)

BLKRL initializes a relatively small number variables in common /RLSYS/ and /LC/. Certain dimensioning variables such as NLCIX and NRLC are initialized to values defined in the Model Assembler Dimensions Module. This is done via common module RLSIZ which is read by the Model Assembler and incorporated into BLKRL by a *CALL. The dimensions thus initialized are listed in Table II-4.

vRLPDS and vIRLD are the principal input verbs for system dynamics data. The main objective of RLPDS that is different from IPDS, the general PDS input verb, is to permit the assignment of statistics collection options to system dynamics variables that are to be stored in PDS datasets. In defining PDS tables, dataset types, and datasets, vRLPDS uses the standard modules DEFTAB, DEFDST, and DEFDSI. Modules RLSASN and RLSSET are used by vRLPDS to organize the data structure for system dynamics statistics collection. vIRLD is used to input definitions of system dynamic delay types, where by type is meant a particular form (1st order, 3rd order, or boxcar, material or information) with particular mean value or for boxcars, a particular distribution of values.

The restart verbs IZRL and SVRL are intended for the input and save status parts of node ZREST in the usual MAWLOGS format. They save and retrieve /RLSYS/ and /LC/ data via the standard MAWLOGS restart file NFREST.

D. LINKAGE MODULES

The system dynamics linkage modules permit the definition of virtually any network of flow paths of interest among system dynamics nodes and subnodes. A flow path may be used by a large or a fine resource category. It may contain many or few links and intermediate processing points. Any delay defined via vIRLD may be assigned to any link. A single node may be in any number of flow paths - as an origin, destination, or intermediate handling point. A set of change modules permits modifying the network by adding or deleting links from existing paths, defining new paths, and replacing one link with another. Delays can be changed by altering the

TABLE II-4. VARIABLES INITIALIZED IN cRLSTZ

VARIABLE	DESCRIPTION
NRLDIM	MAXIMUM NUMBER OF COORDINATES ALLOWED IN RLC IN /LC/
NRLC	LENGTH OF RLC
NLCIX	LENGTH OF LCIX IN /LC/
NRLDPR	LENGTH OF RLDPAR IN /RLSYS/
RLDPAR	DELAY PARAMETERS ARRAY SET TO ZERO
NRLST	MAXIMUM NUMBER OF SYSTEM DYNAMICS STATISTICS ALLOWED IN THIS MODEL
NXRLST	CURRENT NUMBER OF SYSTEM DYNAMICS STATISTICS (= 0)
MMAX	MAXIMUM NUMBER OF UNCLSSES ALLOWED (CLASSES IN MGTCL DATASETS)
RLNNAM	SYSTEM DYNAMIC NODE NAMES ARRAY SET TO BLANK
NRLNMX	SIZE OF RLNNAM IN /RLSYS/ (MAXIMUM NUMBER OF NODE NAMES ALLOWED)
NRDTMX	SIZE OF RLDST IN /RLSYS/, ARRAY OF DATASET TYPES ELIGIBLE FOR INPUT VIA vRLPDS

assignments to links using linkage change modules or by changing forms and parameters using vCHRLD.

The linkage modules are listed in Table II-5.

1. Flow Paths

In the MAWLOGS system dynamics simulation capability, a flow path is the course taken by a resource that is transmitted from one node to another. A flow path is identified by its terminal nodes and any intermediate nodes. The segment between two adjacent nodes in a flow path is referred to as a link. A delay (possibly zero) is associated with the traversal of a link. The status of a link is specified by the rate of flow along it and, if the delay time is positive, by the amount of resource in transit.

Two general kinds of flow paths are recognized - demand and return. More general terms would be send and receive. The important distinction is direction of flow. Send flow paths lead away from the owner of the path. Receive flow paths lead toward the owner. The owner of a flow path is the terminal node that originates demand or is the recipient of returns.

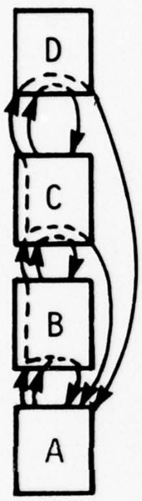
To illustrate the ideas introduced so far, consider Figure II-5. It shows four nodes, each supporting the one immediately below it. In addition, the bottom node, A, receives backup support directly from nodes C and D. The table in the figure identifies all paths, owners, and whether the paths are send or receive paths. Suppose that the return path B→A has a delay of d time units. Then if the flow rate along it is a constant value r, the amount of resource that is in the path is the product rd. If the delay is represented as a probability distribution and the rate r fluctuates, then the amount of resource in transit also fluctuates. But if the average delay and flow are d and r, respectively, then the amount in transit will fluctuate about the product rd. The data structure by which flow paths, delays, and amounts in transit are represented in a MAWLOGS model is described in Chapter III.

TABLE II-5. LINKAGE MODULES

MODULE	DESCRIPTION
	VERBS
vADDLK	ADDS A LINK TO AN EXISTING DEMAND PATH
vADRLK	ADDS A LINK TO AN EXISTING RETURN PATH
vBRLKD	BRANCHES, ONE BY ONE, TO SEVERAL DEMAND PATHS HAVING THE SAME OWNER AND RESOURCE
vBRLKR	BRANCHES, ONE BY ONE, TO SEVERAL RETURN PATHS HAVING THE SAME OWNER AND RESOURCE
vCHDLK	SUBSTITUTES A NEW DEMAND LINK FOR AN EXISTING ONE
vCHRLK	SUBSTITUTE A NEW RETURN LINK FOR AN EXISTING ONE
vDEDLK	DELETES AN EXISTING DEMAND LINK
vDERLK	DELETES AN EXISTING RETURN LINK
vMAPLK	MAPS RATES AND IN-TRANSIT RESOURCES FROM OLD LINKS TO REVISED OR NEW LINKS
vRLDLK	DEFINES A DEMAND LINK
vRLFLK	CONVERTS LINKAGE DEFINITION AND CHANGE DATA FROM SET-UP FORM TO REFERENCE FORM
vRSTAK	PRINTS REPORT SHOWING FLOW NETWORK-PATHS, LINKS, FLOWS, DELAYS, AND RESOURCES IN TRANSIT
vRLRLK	DEFINES A RETURN LINK
	ROUTINES
ADSLC	ADJUSTS THE COUNT OF FLOW PATHS OF PARTICULAR OWNER AND RESOURCE OF WHICH A PARTICULAR NODE IS A MEMBER
ISTAKR	FINDS, IN A PUSHDOWN STACK REPRESENTING A FLOW PATH, THE ENTRY CORRESPONDING TO A SPECIFIED NODE
IXLCIX	ALLOCATES SPACE IN LCIX IN /LC/
LCIXGC	COLLECTS GARBAGE IN LCIX, I.E., RECLAIMS FOR FURTHER USE SPACE THAT WAS OCCUPIED, THEN RELEASED
LRNIX	RETURNS INDEX OF LINKED-REATES DATASET TYPE NAME IN ARRAY RLDST IN /RLSYS/
NEWLCX	USED TO FIND NEW LOCATION IN LCIX AFTER GARBAGE COLLECTION

TABLE II-5. LINKAGE MODULES (CONTINUED)

MODULES	DESCRIPTION
	ROUTINES (CONTINUED)
RELCIX	RELEASES A BLOCK OF LCIX
REVRLS	AFTER DSPOOL GARBAGE COLLECTION, REVISES POINTERS TO VARIABLES IN DSPOOL ON WHICH STATISTICS ARE BEING COLLECTED
RLLNK	STORES LINKAGE DEFINITIONS IN SETUP FORM, ONE BY ONE, AS THEY ARE SPECIFIED BY DEFINITION AND CHANGE MODULES
RSETRS	RESETS LINKAGE POINTERS INTO DTABLE IN /TBLCOM/ AFTER DTABLE GARBAGE COLLECTION
SAVLK	CONVERTS MEMBERSHIP LINKAGE DATA FOR A NODE FROM REFERENCE TO SETUP FORM
TRSTAK	CONVERTS A FLOW PATH FROM REFERENCE TO SETUP FORM



PATH	OWNER	TYPE
A B	A	S
A B	A	R
A B C D	A	S
A C	A	R
A D	A	R
B C	B	S
B C	B	R
C D	C	S
C D	C	R

Figure II-5. Flow Path Examples

2. Definition of Flow Paths

Modules vRLDLK and vRLRLK are used to define demand links and return links, respectively. For example, the reference

RLDLK(P = * A, 3.)

defines a demand link to node A, having a delay of type 3, i.e., whose index among those read in by vIRLD is three. The link is to be made part of the flow path represented by PDS element LKDST(LKCDS). IXDRAT. LKDST, LKCDS, and IXDRAT are in /RLSYS/. If no link has been defined previously for the flow path, the link to A is the first, i.e., the one leading from the owner node itself. If other links have been defined in this path by previously encountered RLDLKs, the link to A is entered as an extension of the existing path. It has proven advantageous, to date, to include the owner node explicitly as the first element of the path. Thus, the sequence

RLDLK(C, 0.), RLDLK(B, 2.), RLDLK(A, 3.)

defines the demand path

C(owner) → D → A

If the path were subsequently altered with the reference RLDLK(F,0.), a link from A to F would be added to the end of the path, resulting in

C(owner) → D → A → F

When a zero delay reference is used, flow on the link is modeled as subject to no delay and, as a consequence, there is zero resource in transit.

The references

RLRLK(C, 1.), RLRLK(B, 2.), RLRLK(F, 1.)

define the return path

$$C(\text{owner}) \leftarrow B \leftarrow F$$

referenced by PDS element LKDST (LKCDS). IXRRAT. A subsequent reference for this path

RLRLK(G, 1.)

results in

$$C(\text{owner}) \leftarrow B \leftarrow F \leftarrow G$$

Thus, for both demand and return paths, the order in which nodes are included in the path determines their proximity to the owner, with the earlier ones nearer.

The values of LKDST, the PDS dataset type containing flow path pointers, and LKCDS, the coordinates of the dataset, will often have been set by vRLOOP. The particular path within the dataset, represented by IXDRAT or, for return paths, IXRRAT, will have been set by the module under whose aegis the links are being defined, or as an adjusted value by vBRLKD or vBRLKR. The latter modules, one for demand paths, one for return paths, step IXDRAT or IXRRAT, respectively, through a number of values specified in the argument NBR of the modules. The initial values are as specified by the host module. Each value of the PDS dataset element index potentially represents a different flow path. For the kth value of the index, PSk of the module is executed. Each PS for which a flow path is to be defined may be filled with a suitable sequence of RLDLKs or RLRLKs. For example, in

Figure II-4 that the return paths from B to A, C to A, and D to A are represented by elements 6, 7 and 8 of LKDST(LKCDS). Then assuming that supporting logic has suitably set LKDST, LKCDS, and IXRRAT (=6), then the pattern

```
BRLKR(P = 3  $
        1 = RLRLK(P = *A, d),   RLRLK(P = *B, d)  $
        2 = RLRLK(P = *A, d),   RLRLK(P = *C, d)  $
        3 = RLRLK(P = *A, d),   RLRLK(P = *D, d) )
```

would suffice. (Lower case d has been used to fill the delay parameter position for convenience. It should also be noted that, as a practical matter, it is often possible to omit the source node of a return flow from the path.)

3. <u>Changes to Flow Paths</u>

It will often be desirable to change the flow path network in the course of a model run to reflect evoluation of the system. Modules to add, change, and delete demand links and return links have been provided for this purpose. For demand links the modules are ADDLK, CHDLK, and DEDLK. For return links similar modules, having an R in place of the middle D, exist. Referring to Figure II-4 again, the reference DEDLK(*C) could be used to delete node C from the demand path A → B → C → D resulting in the path A → B → D. Resources in transit from B to C would be combined with those in the C to D link to form the B to D amount. To include node B in the return path from D to A, use ADRLK (*D, *B, delay). This changes path A ← D to A ← B ← D. (Of course node B must contain logic that insures that the flow from D can be received and transmitted onward into the link from B to A.) To illustrate the idea of substituting one link for an existing one, suppose a new source node E is activated and node C in Figure II-4 is now to be supported by E instead of D. Then CHDLK(*D, *E, delay) could be used to replace the demand link to D with one to E. Resources in transit to D and their flow rate would be transferred to the new link. The delay

assocated with the new link would be that referenced in CHDLK. Thus, the path C→D is changed to C→E.

As with link definition, the flow path affected by the link change modules is the one represented by LKDST(LKCDS). IX_RAT. And, as before, vBRLKD and vBRLKR are convenient modules for addressing multiple paths.

4. Model Control for Linkage Definition and Change

System dynamics flow paths are defined and modified when system dynamics nodes and subnodes are cycled through in setup mode. As was described earlier, a setup pass is automatically executed at the beginning of a cold start of a model. Subsequent setup passes can be scheduled as required. In a setup pass, only the portions of module logic reserved for setup use are executed. For link definition, these portions will generally consist of a setting of LKDST, LKCDS, IXDRAT and IXRRAT followed by execution of a parameter slot into which link definition or change modules are assumed to have been written. Module vSTGBR (see Part 9 of the Catalog) can be used in such a PS to group definitions and changes by time of implementation.

5. Worker Modules for Linkage Definition and Change

The modules discussed to this point are at the user interface level. They do not themselves operate on the linkage data structure. The two principal modules that do are RLLNK and vRLFLK. All definitions and changes are recorded in a temporary form in pushdown stacks by RLLNK. As a final step in a setup pass, after all definitions or changes have been recorded by RLLNK, verb RLFLK is executed to convert the linkage descriptions from their temporary form to their normal form in /LC/ and /TBLCOM/. The temporary form is designed to faciliate and open-ended accumulation of definitions. The permanent or reference form is designed for efficient access and minimum memory occupancy.

Other important modules include SAVLK and TRSTAK, which are used by RLLNK to convert an existing path to temporary form when it is to be changed. Verb MAPLK is used in PS1 of vRLFLK when linkage changes entail the transfer of resources in transit from one link to another.

An overview of the relations among system dynamics linkage modules is shown in Figure II-6. To establish context, the terms nHOST and vHOST are used to indicate system dynamics nodes and verbs executed in a setup pass under control of vRLCTR in the system dynamics control node. After executing all nHOSTS in a system dynamics node cycle, vRLFLK is executed, typically within PS2 of vRLCTR. The role of RLLNK as the common interface between definition and change modules and the data structure is clearly shown.

E. FLOW RATES MODULES

This small group of modules performs the important functions of retrieving or updating flow rates in the system dynamics flow network. Two are concerned with changing rates - UDDR for demand rates and UDRR for return rates. Two are concerned with changing an attribute of a rate. Examples of a rate attribute are density or dollar value of the resource. The first attribute in a RATATT dataset associated with a rate are updated by UDDRA1 for demand rates and UDRRA1 for return rates. RLRAT returns the current value of a specified rate. These modules are listed in Table II-6.

An overview of the linkage and rates modules as they fit into the overall MAWLOGS model control framework is shown in Figure II-7.

F. DELAY MODULES

1. Introduction

The delays experienced by a resource as it flows through a system are usually of considerable interest in assessing system performance. Delay is incurred in such processes as handling, storing, queueing for a processing resource, and travel. In discrete-event MAWLOGS delays are simulated by means of the time file: an entity is put into the time file at the beginning of the process and is scheduled to emerge at the end of the simulated processing time. To simplify the model, a short process may

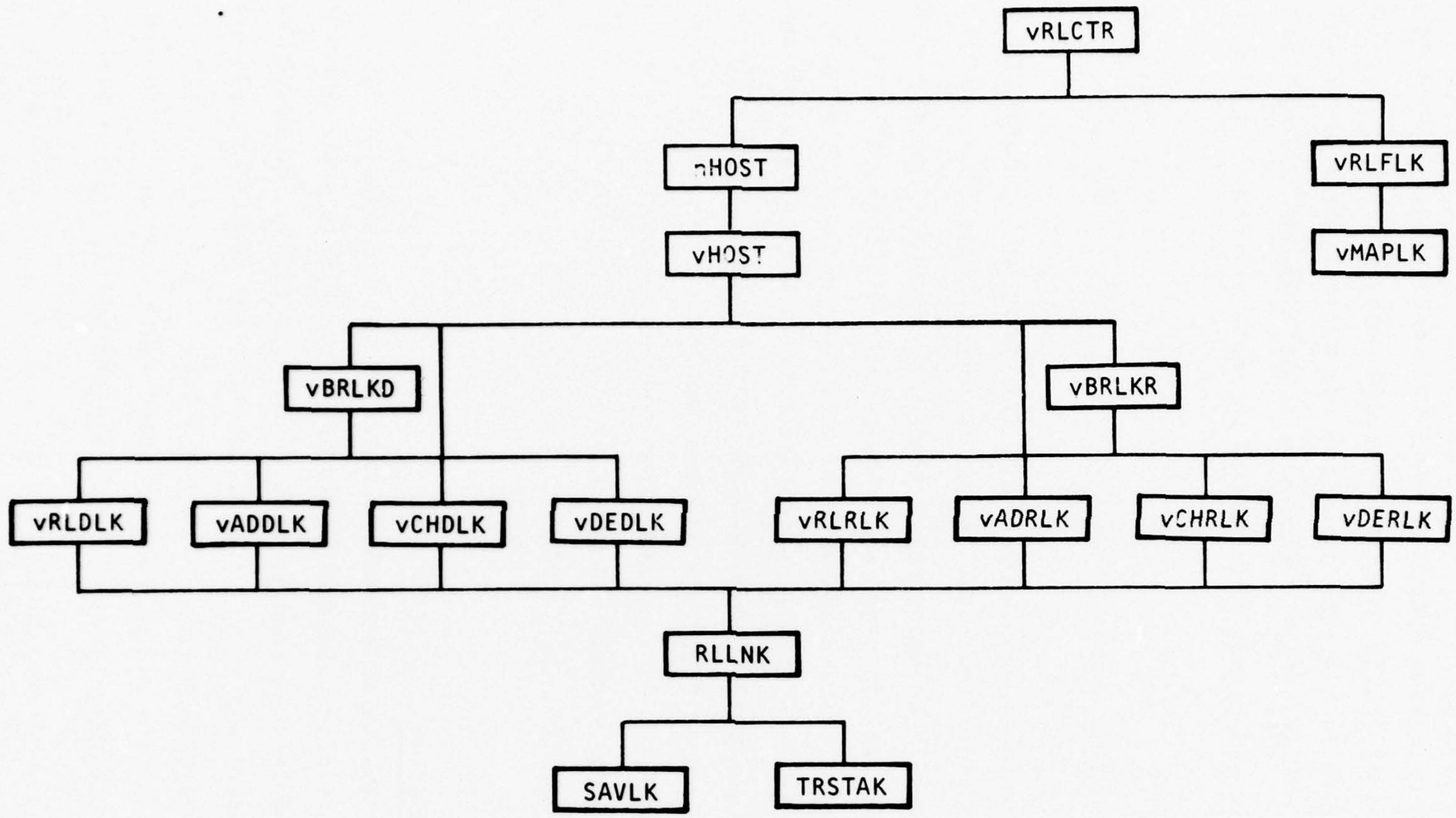


Figure II-6. Schematic of Linkage Module Family

TABLE II-6. FLOW RATE MODULES

MODULE	DESCRIPTION
	VERBS
	NONE
	ROUTINES
RLRAT	RETURNS CURRENT VALUE OF THE FLOW RATE ON A SPECIFIED LINK
UDDR	SETS THE RATE ON A SPECIFIED DEMAND LINK TO A NEW VALUE
UDDRA1	SETS THE FIRST ATTRIBUTE IN THE RATATT DATASET ASSOCIATED WITH A SPECIFIED DEMAND LINK TO A NEW VALUE
UDRR	SETS THE RATE ON A SPECIFIED RETURN LINK TO A NEW VALUE
UDRRA1	SETS THE FIRST ATTRIBUTE IN THE RATATT DATASET ASSOCIATED WITH A SPECIFIED RETURN LINK TO A NEW VALUE

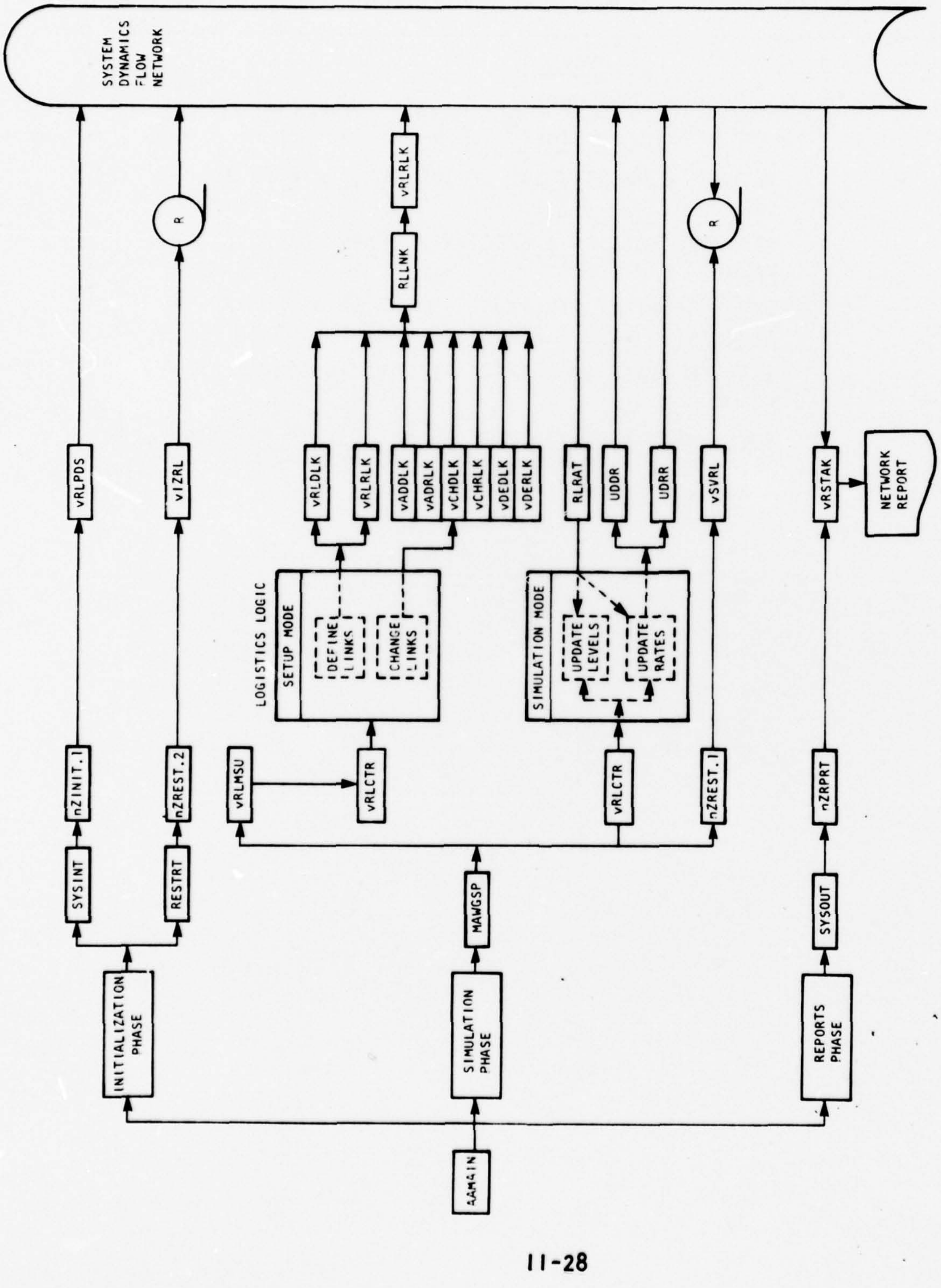


Figure II-7. Schematic of Links and Flow Rates Modules

be simulated as taking zero time while having its time combined with that of a related process with a longer duration. The times associated with particular instances of a process may be drawn at random from a probability function that represents the range of times that characterizes the process.

For continuous flows the time file is not a practical means for simulating delays. It is possible, in principle, to schedule via the time file each time-step-sized increment of flow to be delayed. But the use of the delay methods described below will usually conserve computer memory and execution time. (Use of the time file in this way is equivalent to an infinite-order exponential delay.) The delay forms introduced by Forrester (see Industrial Dynamics) are exponential delays of the 1st and 3rd order, and what he calls "boxcar trains." A boxcar train is a series (train) of cells (boxcars). In practice, each cell is a component of a table that approximates a delay distribution function of arbitrary form.

Consider a first order exponential delay. It represents the process sketched in Figure II-8. The output rate, rout, is equal to ℓ/d, where ℓ is the current level, d is the mean delay. The level changes in a time step, Δt, by the amount

$$\Delta l = (rin - rout)\Delta t.$$

If there were no input rate, the level l would decay from an initial value L at t = o according

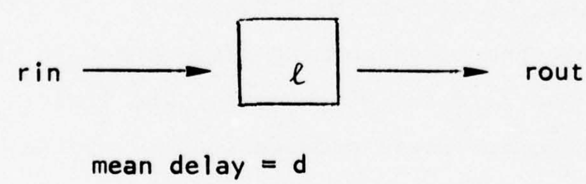


Figure II-8. Model of First Order Exponential Delay

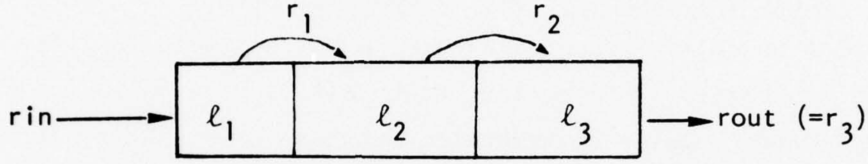
to the function

$$\ell(t) = L\, e^{-t/d}$$

A third order exponential delay is modeled as a sequence of three first order delays, each characterized by an average delay equal to one third of the total delay. This idea is sketched in Figure II-9. If there is no input rate, the total level of resource in the third order delay is given by

$$\left[\ell(t) = Le^{-td} \quad 1 + \frac{t}{d} + 1/2\left(\frac{t}{d}\right)^2\right],\ d = D/3$$

where L is the level at t = o and D is the average total delay. The internal flow rates r_1 and r_2 are given by ℓ_1/d and ℓ_2/d, respectively. The output rate, rout, is ℓ_3/d, where d = D/3.



MEAN DELAY = D

MEAN DELAY OF EACH CELL = D/3

Figure II-9. Model of Third Order Exponential Delay

The outputs from 1st and 3rd order exponential delays for a unit impulse input at time zero are graphed in Figure II-10. For the 1st order delay, 50 percent has passed through after 0.7d time units have elapsed, 80 percent by 1.6d, 95 percent by 3.0d, where d is the mean delay. For a 3rd order delay, the corresponding times are 0.9d for 50 percent, 1.4d for 80 percent, and 2.1d for 95 percent. The limiting case of an exponential delay of infinite order would produce a unit impulse output at time d.

Delays of arbitrary form, called boxcar delays in MAWLOGS system dynamics, are represented as a series of cells, in a manner analogous to 3rd order exponential delays, with each cell operating as a 1st order exponential delay. But in a boxcar delay, the cells may have differing time widths. Also, each cell is assigned a probability that represents the fraction of input flow that incurs the cumulative delay associated with

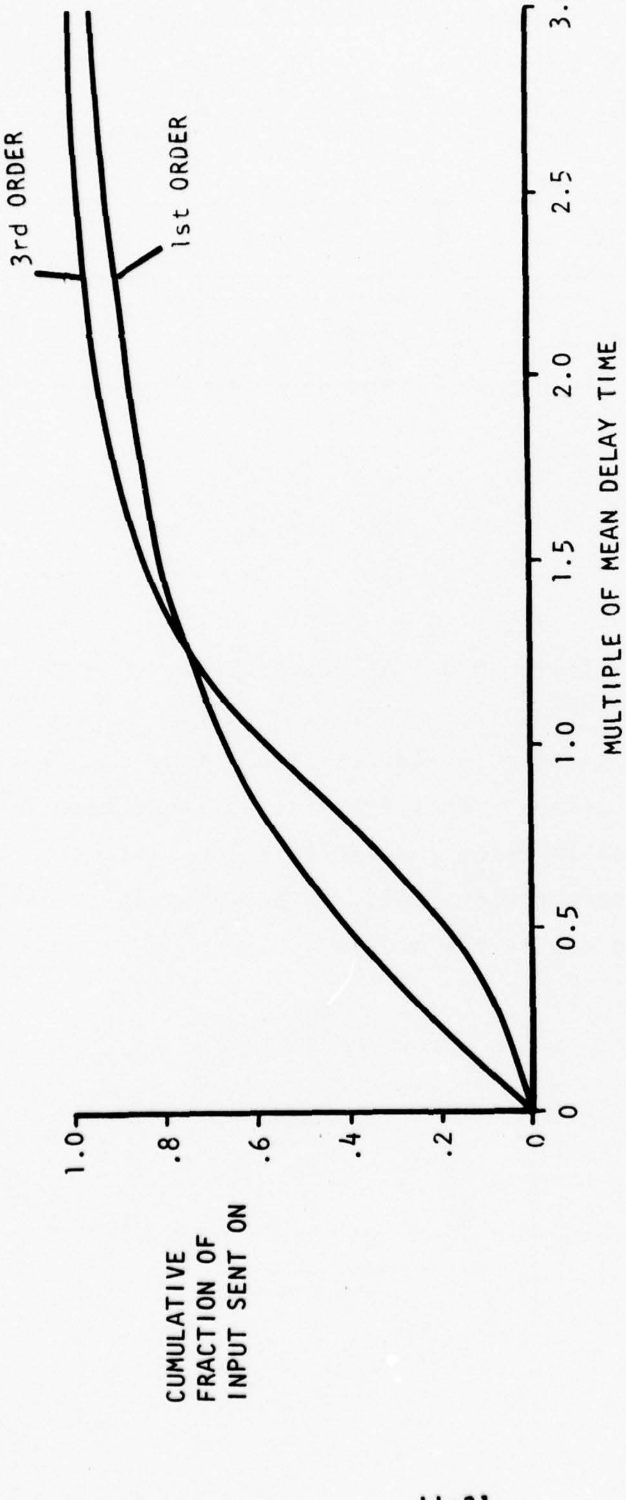
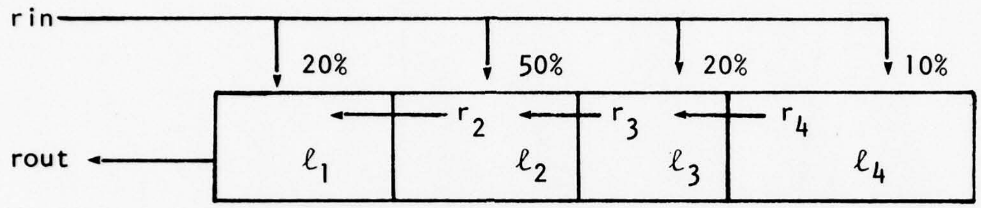


Figure 11-10. Graphs of 1st Order and 3rd Order Exponential Delays (Unit Impulse at t=0)

that cell. To illustrate, consider a delay in which 20 percent of the flow takes 1 day, 50 percent takes 5 days, 20 percent takes 10 days, and 10 percent takes 20 days. Figure II-11 shows the boxcar model for this delay. The output rate of cell i is $r_i = \ell_i/d_i$. The output of the delay is rout = $r_1 = \ell_1/d_1$. The mean delay is $\sum_i p_i \cdot \text{cum } d_i = 7.7$ days.



d =	1	4	5	20
cum d =	1	5	10	30
prob =	.2	.5	.2	.1

Figure II-11. Model of Boxcar Delay

The delay models shown in Figures II-8, II-9, and II-11 are what Forrester calls material delays. They are distinguished from information delays of the same general forms by the explicit identification of both a level and an output rate for each cell. In an information delay, only the level of each cell is explicitly modeled.

2. MAWLOGS Implementation

As is described in detail in the next chapter, system dynamics delay data are organized into two basic categories. One consists of descriptors of delays. The descriptors of a delay are its form (1st order exponential, 3rd order exponential, or boxcar, information or material) its mean if exponential or its table of probabilities and cell widths if boxcar, and a nomenclature. A delay so characterized is identified by an index number. The second category of delay data consists of the PDS datasets of type RLD in which particular implementations of the delays - the cells, levels, and internal rates - are stored. A generic delay, i.e., one characterized by a form and a parameter or distribution table, may be applied in any number of places in a model. Each application is represented by a unique RLD dataset which reflects the input history peculiar to that application. For example, a generic delay might represent the time required to process a particular type of resource through a particular type of handling. If there are ten places in the model where that type of resource is subjected to that type of handling, there would be ten RLD datasets, all governed by the form and parameter of a single generic delay, but having different status to the degree that their input flows differed in volume or time pattern.

3. Modules

The MAWLOGS system dynamics delay modules are listed in Table II-7. They consist of an input module, vIRLD, two output modules, vORLD and vDLEVS, a set of accessing modules centered on RLDLY, and a set of change modules managed by vCHRLD, the input module for changes. Restart data for delays are handled by the overall system dynamics and PDS modules for restart, vIZRL, vIZPDS, vSVRL, and vSVPDS. This pattern of relations is shown in Figure II-12.

Module RLDLY is the intended interface between user logic and system dynamics delays. It employs one of the six modules corresponding to one of the six delay forms as appropriate to update the levels and rates within the delay and to obtain the new output rate. Proper maintenance of

TABLE II-7. DELAY MODULES

MODULE	DESCRIPTION
	VERBS
vCHRLD	READS CHANGES TO DELAY PARAMETERS
vDLEVS	PRINTS CURRENT RESOURCE CONTENT OF EACH DELAY
vIRLD	READ DEFINITIONS OF DELAYS-FORMS AND PARAMETERS
vORLD	PRINTS DELAY DESCRIPTIONS-FORMS AND PARAMETERS
	ROUTINES
ARLDIX	ASSIGNS AN INDEX AS COORDINATE FOR A NEW INSTANCE OF A SYSTEM DYNAMICS DELAY, AND CREATES AN RLD DATASET IN WHICH TO STORE ITS CELLS
DLYLEV	RETURNS THE RESOURCE CONTENT OF A DELAY
DMAP	TRANSFERS A FRACTION OF THE RESOURCE CONTENT OF ONE DELAY INTO ANOTHER
DTIM	CONSTRUCTS TABLE OF END TIMES OF CELLS IN A DELAY
RLDLY	PRINCIPAL INTERFACE WITH USER MODULES-RETURNS OUTPUT RATE OF A SPECIFIED DELAY AFTER UPDATING IT FOR THE CURRENT TIME STEP
RLDREV	REVISE LEVELS IN A DELAY TO REFLECT NEW PARAMETERS
RLDRPT	PRINTS DELAY DESCRIPTIONS-FORMS AND PARAMETERS
RLIDB	BOXCAR INFORMATION DELAY-UPDATE AND RETURN OUTPUT VALUE
RLID1	1ST ORDER EXPONENTIAL INFORMATION DELAY-UPDATE AND RETURN OUTPUT VALUE
RLID3	3RD ORDER EXPONENTIAL INFORMATION DELAY-UPDATE AND RETURN OUTPUT VALUE
RLMDB	BOXCAR MATERIAL DELAY-UPDATE AND RETURN OUTPUT VALUE
RLMD1	1ST ORDER EXPONENTIAL MATERIAL DELAY-UPDATE AND RETURN OUTPUT VALUE
RLMD3	3RD ORDER EXPONENTIAL MATERIAL DELAY-UPDATE AND RETURN OUTPUT VALUE
RSETRD	RESET POINTERS OF BOXCAR DELAYS INTO DTABLE AFTER DTABLE GARBAGE COLLECTION (DESCRIBED IN PART 9 OF THE MODULE CATALOG)

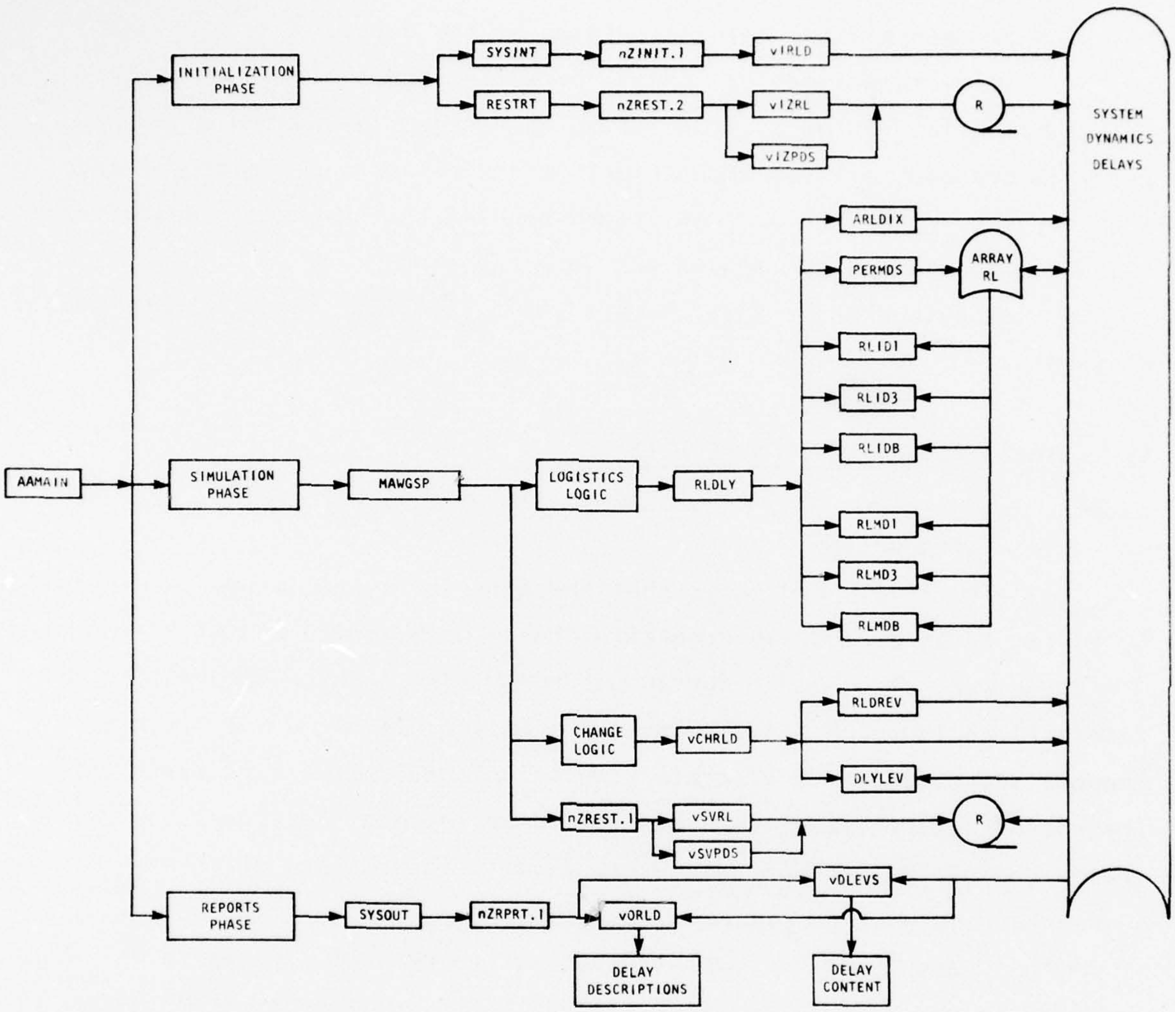


Figure II-12. Schematic of Delay Modules

a system dynamics delay requires one and only one update of an RLD dataset per time step. If a delay is being accessed for the first time, RLDLY will call ARLDIX to assign an identifying coordinate and create an RLD dataset in which to maintain the internal status of the delay.

The change module, vCHRLD, permits changes of the parameter (or distribution table) and form of any generic delay. When a generic delay is changed, all implementations of the delay, i.e., all cell sizes and statuses in the RLD datasets corresponding to the generic delay, are revised. Module RLDREV is used for such revisions.

Module DMAP is the delays adjunct to vNODMP and vMAPLK discussed elsewhere.

G. STATISTICS COLLECTION MODULES

1. Introduction

The levels and rates that characterize system elements simulated by system dynamics modules generally change only at the end of a time step. (The structure of combined continuous and discrete MAWLOGS models permits interactions in which a system dynamics rate or level is subjected to changes introduced by a discrete simulation module at a time other than the end of a time step. Care must be taken by model designers to insure that such interactions do not overly distort the results of the simulation. A powerful, general procedure for managing such interactions is provided in GASP IV[6] and could be incorporated, at the cost of a potentially large increase in execution time, into the MAWLOGS simulation control system.) Thus, in the terms of the standard MAWLOGS statistics collection scheme described in Part 1.1 of the Module Catalog, system dynamics variables are subject to the TMST form of observation. That is, the statistical measure of interest for a variable is a time-weighted average.

---

[6] Pritsker, A. Alan B., The GASP IV Simulation Language, New York, NY, John Wiley and Sons, 1974.

2. Implementation

A principal goal of the implementation of system dynamics statistics collection in MAWLOGS was that the data be amenable to summarizing and graphing by the MAWLOGS Output Data Postprocessor System (ODPS). On the other hand, it seemed desirable to avoid direct use of the STAT 1 approach to statistics collection in an attempt to capitalize on the potential for reducing central memory requirements inherent in the fact that values change only at the end of a time step. A third objective was to provide for the flagging of which variables to observe and in what form at data input time with the same degree of flexibility as was provided in discrete MAWLOGS.

The method adopted only partially achieved those objectives. It is flexible, compatible with the ODPS, and avoids the need to build calls to STAT 1 into user modules. But, as currently organized, it has not taken advantage of opportunities to conserve central memory (or peripheral memory, either). Further, it does not permit observation of flow rates and levels in links of the network of flow paths. Only variables whose values are stored in PDS datasets can be observed.

System dynamics statistics are collected by module RLSTAT, whose executions are scheduled as a GASP event. An execution of RLSTAT entails the observation of all variables flagged, in the input data, for observation at a frequency whose observation times include the current time step. Nine observation frequencies are recognized - a basic one, usually defined as every time step, and eight corresponding to power-of-two multiples of the basic interval. The kth frequency is one observation per (2^{k-1}) (basic interval) time steps, $k = 1, 2, \ldots, 9$. Thus, in general, a series of observations on a variable constitutes a time sample. The form of observation may be COLCT, HISTO, TACTN, or PRRLST. The first three forms are the standard, discrete MAWLOGS forms. The fourth, PRRLST, consists of printing the current value of a variable. Combinations of TACTN with another form are not currently provided for.

3. Modules

The MAWLOGS system dynamics statistics collection modules are listed in Table II-8. An overview of their interrelations and their position in the overall MAWLOGS model control scheme is shown in Figure II-13.

Variables are flagged for observation in the vRLPDS input data. The form and frequency of collection are indicated. The basic collection interval is also input via vRLPDS and converted to the nearest integer number of time steps. A default value of one time step is used if the data field is blank or less than TSTEP. The form and frequency of collection may be changed or new variables may be added via the inputs of vCHPDS. The flagging scheme is the same as that in vRLPDS. For restart, the general system dynamics modules vIZRL and vSVRL and the general statistics modules (from discrete MAWLOGS) vIZSTA and vSVSTA are used.

Statistics are collected by RLSTAT, which uses modules PRRLST, COLCT, HISTO, and TACTN to record observations in the desired form. RLSTAT is scheduled for execution via the time file as a GASP event. The values it observes are read directly from their DSPOOL storage locations.

The modules ABOXST, DBOXST, and COMPRB bear special attention. They are the only system dynamics statistics modules that need to be referenced in logistics or other user modules. Their purpose is to permit the measurement of elapsed times associated with the flow of a resource through several links and nodes, each with its own delay characteristics. In discrete MAWLOGS, such statistics are readily obtained by carrying a start time with a transaction or shipment as it makes its way through the system, and subtracting from TNOW whenever desired. But the continuous flows through delays that characterize the progress of a resource through the system in a system dynamics model do not permit data-flagging and direct observation of discrete resource packets. The procedure embodied in ABOXST and DBOXST is to reserve a series of time cells in a PDS dataset of type OBOX. The cells are of equal time-width, TW. Their cumulative end points represent times TW, 2TW, 3TW, . . . , nTW into the past, where n is the number of cells. When a quantity of resource enters a flow whose elapsed time is

TABLE II-8. STATISTICS COLLECTION MODULES

MODULE	DESCRIPTION
	VERBS
vIBXST	INITIALIZES BOXCAR TRAIN DATASETS OF TYPE OBOX USED FOR MEASURING TIME DELAYS
vRLPDS	READS STATISTICS COLLECTION CODES AS PART OF PDS INPUT DATA FOR SYSTEM DYNAMICS VARIABLES
	ROUTINES
ABOXST	ADDS RESOURCES TO OBOX DATASET BEING USED TO MEASURE TIME DELAYS
CHRLST	CHANGES COLLECTION TYPE AND FREQUENCY FOR SYSTEM DYNAMICS VARIABLES
COMPRB	COMPRESSES THE DATA IN AN OBOX DATASET WHEN CELL WIDTHS MUST BE INCREASED TO ACCOMMODATE LONGER TIME DELAYS
DBOXST	REMOVES RESOURCES FROM AN OBOX DATASET AND RETURNS THE AVERAGE ELAPSED TIME SINCE THE REMOVED RESOURCES WERE ENTERED INTO THE DATASET
PRRLST	PRINTS IDENTIFIES AND CURRENT VALUE OF A SYSTEM DYNAMICS VARIABLE
RLSASN	ASSIGNS TYPE AND FREQUENCY CODES TO A SYSTEM DYNAMICS VARIABLE WHEN FLAGGED IN vRLPDS INPUT
RLSRPT	LISTS SYSTEM DYNAMICS VARIABLES SUBJECT TO STATISTICS COLLECTION AND THEIR FREQUENCY AND COLLECTION TYPE CODES
RLSSET	SORTS STATISTICS POINTERS IN KRLST INTO DESCENDING FREQUENCY OF COLLECTION ORDER
RLSTAT	PRINCIPAL COLLECTION MODULE - CALLS COLCT, HISTO, TACTN, AND PRRLST AS APPROPRIATE FOR EACH VARIABLE WHEN EXECUTED AS A GASP EVENT, THEN SCHEDULES ITS NEXT OCCURENCE

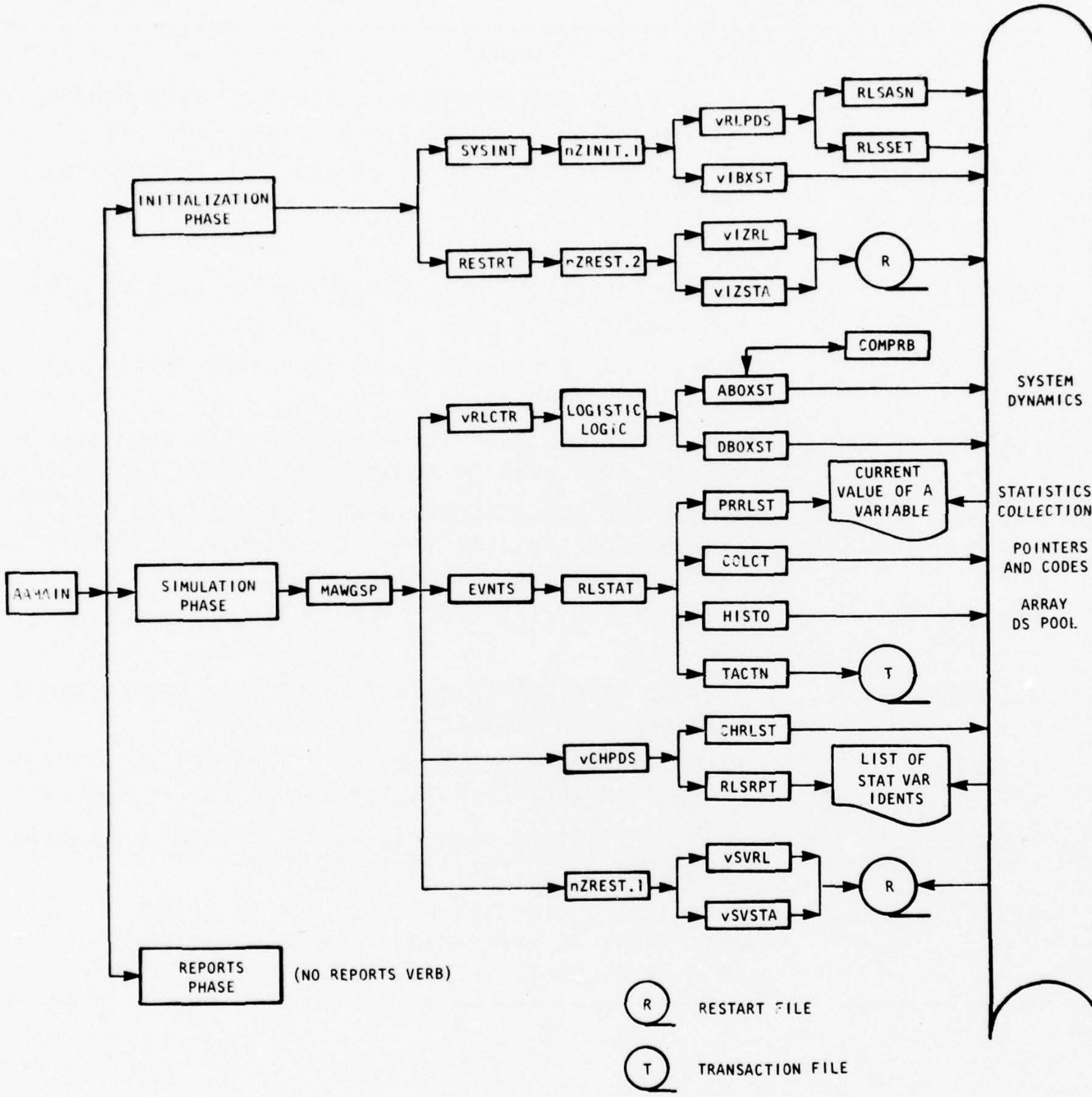


Figure II-13. Schematic of Statistics Collection Modules

to be measured, the same quantity is entered into the youngest cell of the OBOX dataset. When a quantity of resource emerges from the flow path of interest, DBOXST is used to remove an equal quantity from the oldest non-empty cell. The average elapsed flow time associated with the quantity of resource removed is approximately kTW, where k is the number of cells currently nonempty (more refined estimates are made in actuality, reflecting the possibility that TW may encompass several times steps, and that resources are not removed from a cell in equal increments). COMPRB is called by ABOXST when no current cell exists, i.e., when the range of elapsed times being observed exceeds nTW, and hence the youngest cell cannot include TNOW, the current simulation time. COMPRB calculates a new, large TW, and redistributes the resources. The new TW is chosen to be just large enough to create a new empty cell. While the distribution of elapsed times obtained by this means is not reliable, the average time is. Sensitivity to detect changes in elapsed times can be made quite sharp by choosing a large enough number of cells.

H. NODE MAPPING MODULES

Module vNODMP provides a capability for specifying that rates and levels currently associated with one set of nodes in their current proportions be reassigned to another set of nodes in the same or different proportions. Nodes in the second set may be nodes from the first set. The mapping is read by vNODMP during a setup pass. Modules vMAPLK and DMAP use the node mapping to redistribute network flows when a node mapping has been specified. Module vRELNM is available to release the current NODMAP datasets after the mapping has been completed.

I. GENERATION OF EXOGENOUS DEMAND FLOWS

Module vRLDGN is available for this purpose. Arbitrary patterns of flow may be generated. During setup mode, vRLDGN generates flow links for the owner dataset type and element specified as arguments of vRLDGN.

CHAPTER III
DATA STRUCTURE

A. OVERVIEW

The data structures used by the MAWLOGS system dynamics service modules are concentrated in labeled common areas /RLSYS/ and /LC/ created especially for them. Considerable use is also made of /TBLCOM/ and the pushdown stacks described in Part 1 of the Module Catalog as portions of the original MAWLOGS system. Certain PDS dataset types are also part of the MAWLOGS system dynamics data structure. They include RLD, LDSC, OBOX, RATATT, and NODMP. Finally, miscellaneous additions have been made to labeled common areas that were part of discrete MAWLOGS. For example, RTRACE was added to /ZMAWSY/ and RLSNAM was added to /STNAM/.

The data structure will be described from the vantage point of the several functional subfamilies of modules identified in Chapter II, namely Model Control, Flow Path Network, Delays, Statistics Collection, and Node Mapping. The variables in commons /LC/ and /RLSYS/ are defined in tables at the end of the chapter.

B. DATA STRUCTURE FOR MODEL CONTROL

Key variables for model control are TSTEP, the time step by which simulation time is advanced in the system dynamics portion of the model, and RLSW, which is assigned the value LEVELS for levels updating, RATES for rates updating, and NEXT when neither a levels or rates update pass is in progress. RLMODE is used to designate whether a cycle through system dynamics nodes is for rates and levels updating, for which it is assigned value SIM, or for definition or modification of the flow path network, for which it is assigned value SETUPM. Variables RLNOD and KRLNOD contain the system dynamics node number in floating point and integer form, respectively, of the node currently being executed. It is incumbent on the model designer

to begin every system dynamics node with vRLNOD, which sets these variables and, on the initial setup pass through the node cycle, builds table RLNNAM in which the name of node number KRLNOD is stored at RLNNAM (KRLNOD). (This then permits flow links to be defined in the model description in terms of node names, which are converted to numbers by a function that scans RLNNAM.) The dimension of RLNNAM must be specified as the value for *NRLNMX in the Model Assembler dimensions module.

Variable NSTEP is a counter showing the number of time steps that have been executed in the model run so far. When compared with NSTW, the number of time steps to be used for accelerated warmup of flow delays, it controls when the normal updating of delay status begins.

All of the Model Control variables discussed above are in common /RLSYS/.

A variable for causing a logic trace to be printed for a single system dynamics node during model checkout is RTRACE in /ZMAWSY/. It is used as an adjunct to the KTRACE event of discrete MAWLOGS, GASP event number -2. When the third attribute of a KTRACE activation event is nonzero, it is interpreted as a system dynamics node number to be traced and is set into RTRACE. When vRLNOD detects a positive RTRACE, it insures that the KTRACE switch is on, which in turn causes execution of the KTRACE print statements in the modules executed in the node. All nodes may be traced in the usual way - merely by scheduling a KTRACE event. The option of restricting the trace to a single system dynamics node is valuable when there are many nodes and a full trace would be quite voluminous if repeated for many time steps.

C. FLOW NETWORK DATA STRUCTURE

Three data structures will be discussed in this section. The reference data structure will be discussed first. It is the "normal" representation of the flow network, i.e., the one accessed and updated during system simulation. Two temporary data structures will then be described. The first is used during initial definition of the network. The second is an extension

of the first that permits selective changes to be made after the initial definition has been converted to reference form.

1. Reference Data Structure

A flow path is represented by a sequence of contiguous entries in DTABLE in /TBLCOM/. There is one entry for each node in the path. The entry for a node contains the system dynamics node number and a delay coordinate packed into DTABLE(1,*), and the current rate entering the node in DTABLE(2,*). The delay coordinate specifies the delay on the link that leads to the node from its immediate predecessor in DTABLE. A zero delay coordinate represents zero delay. The packing of node number and delay coordinate is by IPAK2.

The direction of flow in a path is downward in DTABLE for demand (or send) flows, upward for return (or receive) flows. Figure III-1 illustrates a send flow path and a receive flow path in DTABLE. They correspond to paths shown in Figure II-5. The numbers to the left of the rectangles

9	A	0	r(→A)
10	B	3	r(A→B)
11	C	7	r(B→C)
12	D	8	r(C→D)

19	A	4	r(D→A)
20	D	0	

Figure III-1. Flow Paths in DTABLE

are postulated locations in DTABLE. The arrows show direction of flow. Node names are used in place of node numbers in the figure to make it easier to relate to Figure II-5. Note that the demand flow into A has no origin indicated. This is typical of flows with an exogenous origin. The flow from A to B passes through the delay represented in dataset RLD(3.), that from B to C through RLD(7.), and from C to D through RLD(8.). The return flow from D to A experiences the delay represented by RLD(4.). Normally, the node of origin of a return flow need not be explicitly present in the path. But an entry for D has been shown in Figure III-1 to more clearly illustrate the upward or backward direction in DTABLE associated with return paths.

A flow path is associated with an owner node and a resource by means of a pointer in a dataset whose coordinates are node and resource. The pointer word is assumed packed by IPAK2 with the number of nodes in the path as the left element and the index of the element having the smallest DTABLE index as the right element. For example, dataset type FCRAT (node, resource) is used in the continuous supply family (Part 7 of the Module Catalog) to associate flow paths with owner nodes. The first element of the dataset is the number of demand paths it can own, ND; the second element is the number of return paths, NR. The next ND elements are pointers to the demand paths in DTABLE. The last NR elements are pointers to return paths. Assume that the demand path shown in Figure III-1 is owned by node A, pertains to resource X, and is identified in the link definition modules as the first demand path of the ND possible paths in an FCRAT dataset. Then the third element of FCRAT(A,X) would contain the pair (4,9). Similarly, if the return path shown in Figure III-1 is the NRth, FCRAT(A,X). (ND+NR+2) would contain the pair (2,19).

An important activity at a node like B, C, or D in Figure II-5 is to scan the incoming demands from all of the nodes it supports. (While only node A is shown in the figure, it could be just one of several, supported by B, C, and D in the same pattern.) This requires that node B, for example, be able to identify all flow paths of which it is a member. This is done by the structure shown in Figure III-2. Every node that has membership in at least one flow path has created for it an LDSC dataset. The LDSC dataset has as its only element a pointer into DTABLE. The pointer, consisting of an element count and position of first element, identifies a list of linked-rate PDS dataset type names. FCRAT is an example of such a name. EXRAT is another used in continuous supply. The DTABLE entries contain a dataset type name in the 1st element and an associated pointer into LCIX in /LC/ in the second. The entries in LCIX are themselves pointers to individual entries in the rate link coordinates array in /LC/, named RLC. The interpretation of these connections is that the node to which the LDSC dataset belongs is a member of one or more of the flow paths owned via LKDST(RLC(*,LCIX)) for all LKDSTs pointed to by the LDSC, and for all

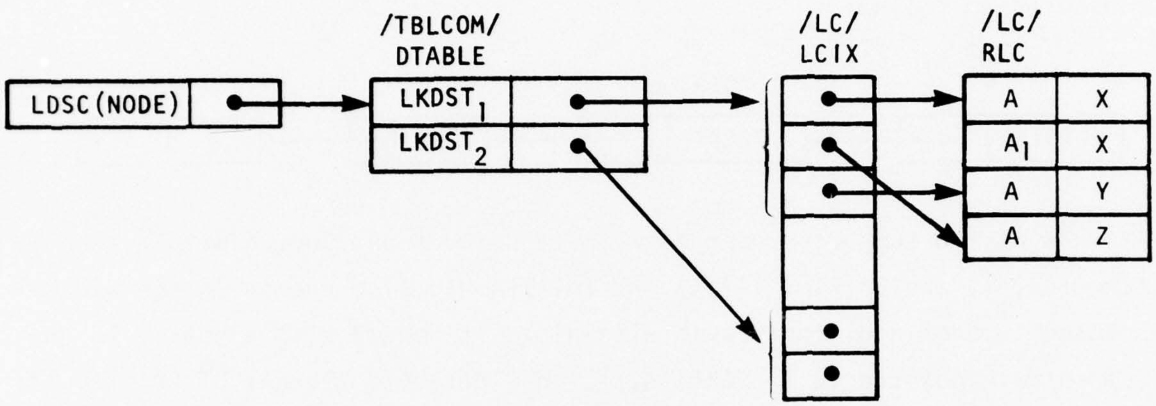
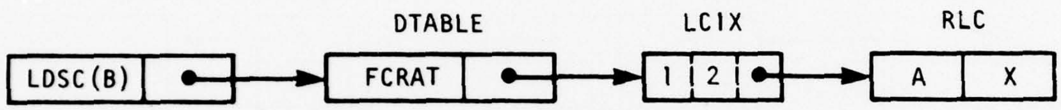


Figure III-2. Member Node Access to Flow Paths

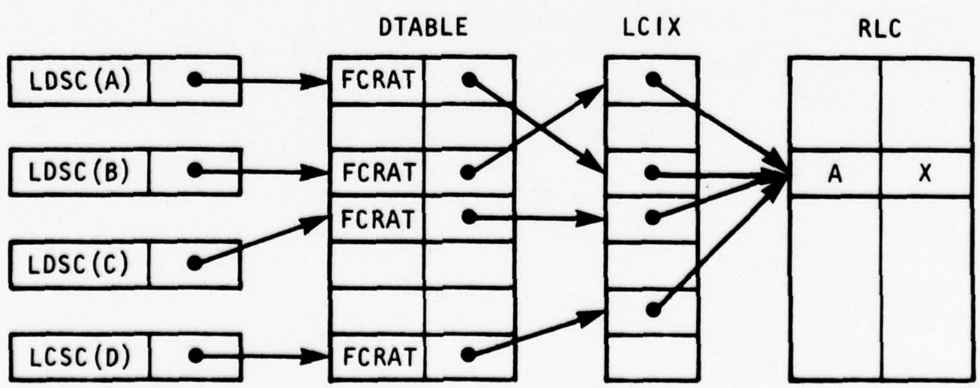
LCIXs pointed to from the LKDSTs. The actual content of LCIX is three packed elements (NREFS, NCDS, IRLC), where IRLC is the position of the coordinates in RLC, NCDS is the number of coordinates, and NREFS is the number of appearances of the LDSC node in the flow paths owned by LKDST(RLC(*,IRLC)). The packing is that done by IPAK2(IPAK2(NREFS,NCDS),IRLC).

To illustrate, consider node B in the example discussed above. Its membership in an FCRAT demand path for node A and resource X would be indicated as follows:



If node B also supported node A1 with resource X and node A with a second resource, Y, the entries (A1,X) and (A,Y) would also appear in RLC and be pointed to from additional LCIX elements. If node B also appeared in one return path pointed to by FCRAT(A,X), the left-most element of LCIX would be 2.

A set of coordinates in RLC may be pointed to by several entries in LCIX. Indeed, there will be as many pointers in LCIX for an entry in RLC as there are distinct nodes in all the flow paths owned by datasets having the coordinates. To illustrate, nodes A, C, and D are also members of the flow path being considered. That single path would lead to the following structure:



To update the rate for a particular node in a particular flow path, modules UDDR and UDRR require as arguments a node number and a flow path pointer from the owner dataset.

The module vRLOOP was designed to find all owner datasets of a specified dataset type in whose paths the current node (i.e., RLNOD in /RLSYS/) appears as a member at least once. The dataset type name is specified as an argument of vRLOOP. vRLOOP traverses the structure emanating from LDSC(RLNOD) to find the datasets. For each dataset found, its name is set into FCRTYP in /SUPC/, the index of its coordinates in RLC is set into IXRLC in /SUPC/, and its elements are set into RATPS in /SUPC/. (/SUPC/ is the continuous supply common. Since the function of vRLOOP is of much wider application, it would be desirable to move FCRTYP, IXRLC, and RATPS to /RLSYS/.) Then PS1 is executed for the dataset.

2. Temporary Form for Initial Definition

The initial definition data structure represents the flow network in essentially the same terms as the reference data structure. The principal difference is that the lists that are stored in segments of DTABLE and LCIX in the reference structure are stored in pushdown stacks, so that their sizes can be extended whenever necessary as the flow paths are built up. An LDSC dataset is created whenever a new node enters the network. The RLC table of coordinates is extended whenever a new set of coordinates is encountered.

During the definition phase, the pointers to flow paths are not stored directly in the owner dataset. Instead, the pointers in the datasets are negative. The absolute value of a negative pointer is the location in the pushdown stacks array, IZHOLD, of a pointer entry. The IZK element (i.e., the second word) of the pointer entry is itself a negative pointer to the top of a pushdown stack in which the flow path is being built. This structure is illustrated in Figure III-3. Note that IZKC in the flow path entries contains node number and delay type code. The delay type code is an index that identifies a particular combination of delay form and parameters. RLD coordinates are assigned during conversion of the flow paths to reference form.

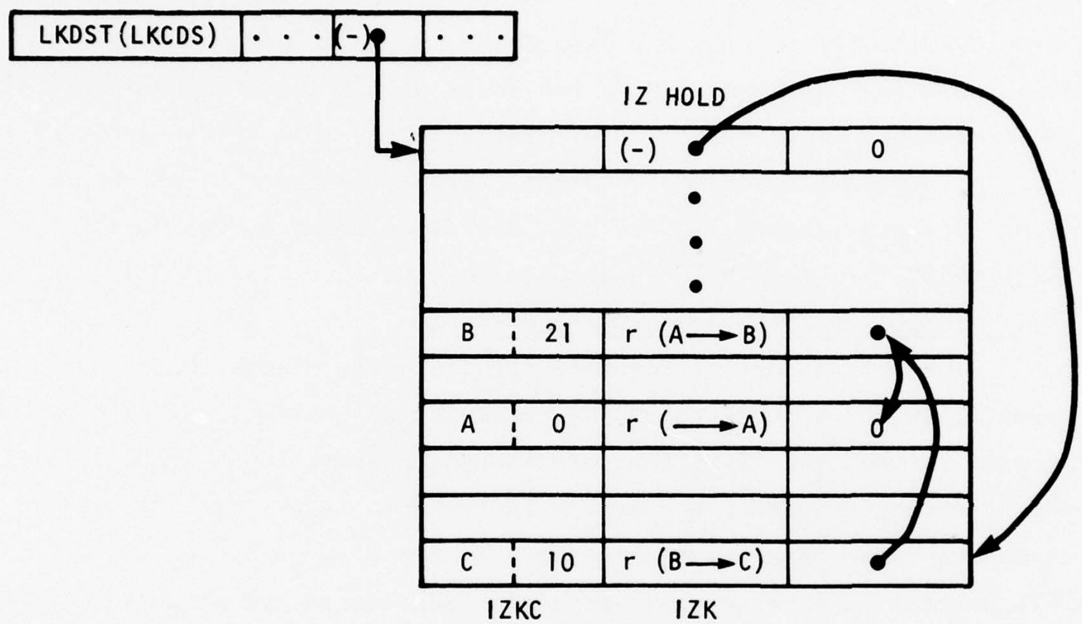


Figure III-3. Flow Path During Initial Definition

The flow path illustrated in Figure III-3 is the A → B → C → D demand path used for illustration above and sketch in Figure II-5. At the stage shown in Figure III-3, node D has not yet been defined as a member of the path. When it is defined (by a reference to vRLDLK), it will be added to the top of the pushdown stack, and the pointer in LKDST(coords) will be set equal to the negative index of the position of the new entry in IZHOLD.

The access path for a member node is built in a manner that parallels the reference form. The first time a node is defined as a member of a flow path, an LDSC dataset is created, and a pushdown stack of dataset type names is initiated in IZHOLD. The LDSC element is set equal to the negative of the index in IZHOLD of the dataset types stack. At the same time, the coordinates of the path owner dataset are added to RLC in /LC/ if they are not already present, and a pushdown stack of pointers into RLC is initiated. The top of this stack, which corresponds to a segment of LCIX in the reference structure, is stored as a positive IZHOLD index in the IZK (i.e., second) element of the stack entry containing the dataset type name. The stacks of names and RLC pointers are expanded as the LDSC node is designated a member of paths having other owner datasets. This structure is sketched in Figure III-4.

Links are defined by modules vRLDLK and vRLRLK. Arguments of the modules specify the node into which the rate on the new link will flow, and the delay that will be incurred. These values are passed in turn as arguments of RLLNK, the module that actually constructs the data structures illustrated in Figures III-3 and III-4. RLLNK assumes that the owner dataset type and its coordinates are the current values of LKDST and LKCDS, respectively, in RLSYS. The particular flow path of the potentially many that can be pointed to from a single owner dataset is received by RLLNK as argument IXRATE, the index of the corresponding pointer element in LKDST (LKCDS). An effective way of setting LKDST(LKCDS) for an owner node is to:

(1) Input via vRLPDS all rate-owning datasets for the model
(2) Use vRLOOP at nodes for which such datasets were established to process each dataset in turn and define flow paths

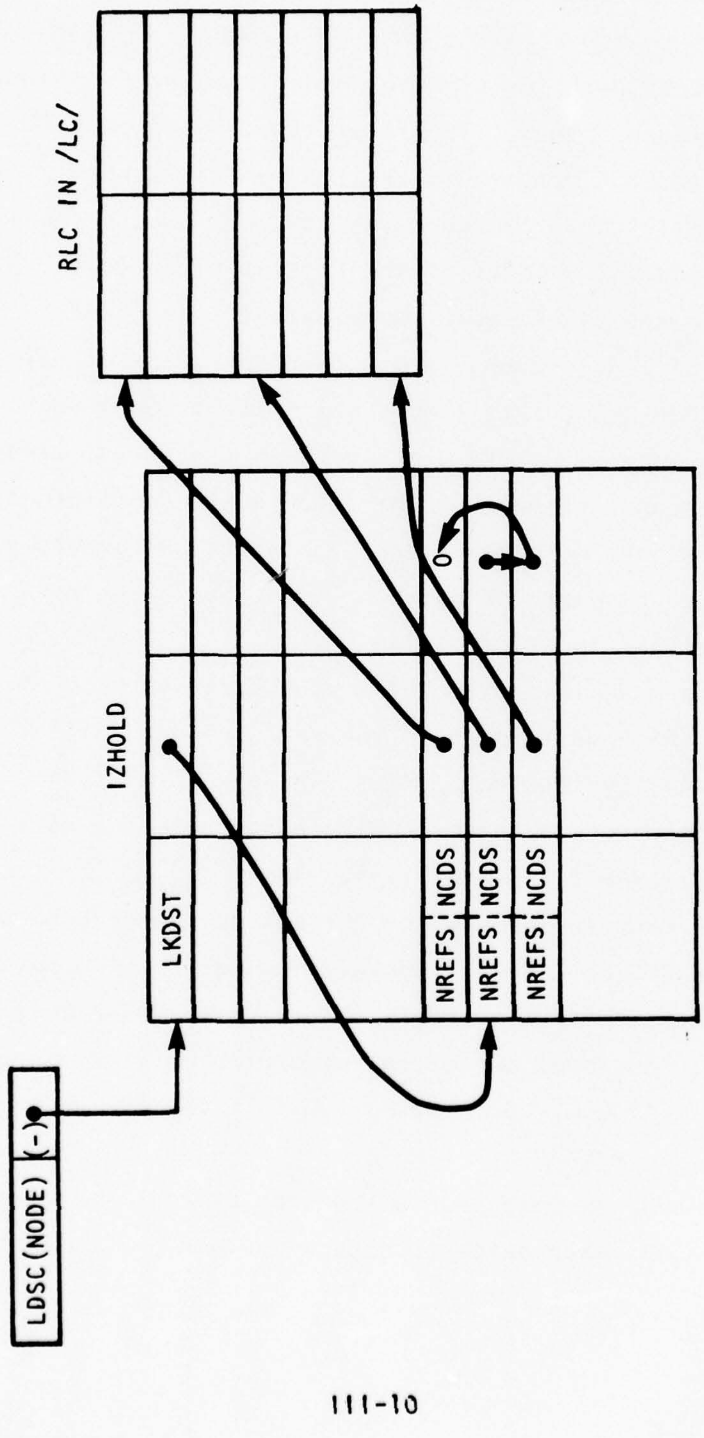


Figure III-4. Flow Path Membership in Initial Definition

(3) When multiple paths are to be defined, use vBRLKD or vBRLKR to index through the elements of the owner dataset.

At the end of the first setup pass, after all flow paths have been defined, vRLFLK is used to convert the pushdown stacks to segments of DTABLE and LCIX. The IZHOLD space is released and the various pointers are adjusted to correspond to the new locations and the conventions of the reference form.

3. Changes to the Flow Path Network

The procedure used to modify the reference form of the flow path network entails conversion of those segments to be changed back to a form similar to the one used for initial definition, and then attaching the change information in extensions of the form. Thus, a path to be modified is removed from DTABLE and put into a pushdown stack. Also, the access structure for member nodes affected by the change is converted to initial definition form, i.e., lists dataset type names in DTABLE and their associated LCIX pointer lists are converted to pushdown stacks. Their former DTABLE and LCIX space is released for reclamation when the garbage collection procedures for those arrays are executed.

The converted membership data structure is identical to that portrayed in Figure III-4. But differences occur in the data structure for flow paths. The pushdown stack containing the "old" flow path is pointed to from IZKC instead of IZK of the pointer entry in IZHOLD. The pointer entry itself is pointed to by a negative IZHOLD index in the owner dataset, as shown in Figure III-3. Also, allowance is now made for the possibility that the flow in a link may have associated with it attributes stored in a RATATT dataset. When this is true, the IZK element of the pushdown stack entry, which normally contains the current flow rate, is changed to a negative index of a two-entry pushdown stack containing the rate and number of RATATT attributes in the top entry and the RATATT attribute values themselves in the bottom entry. The RATATT dataset is released and its elements are saved in a pushdown stack this way because it is known that the coordinate of the RATATT dataset, namely the location in DTABLE of

the rate it characterizes, will be different as a result of the temporary removal of the flow path from DTABLE and release of its DTABLE space.

Changes are initially recorded in a pushdown stack pointed to from -IZK (second element) of the pointer entry in IZHOLD whose IZKC (first) element contains the negative index in IZHOLD of the old flow path. An entry in a change stack contains a change code in the IZKC position and a pointer to a two-member stack in the IZK position. The content of the two elements is:

new node	d	r new	•
old node	d	r old	0
IZKC		IZK	

Four change codes are recognized:

- -1 delete an oldnode
- -2 insert newnode after oldnode
- -3 substitute newnode for oldnode
- -4 insert newnode before oldnode.

Thus, a new node and an old node are specified in most change notices. The terms "before" and "after" are interpreted relative to the direction of flow.

Figure III-5 is a sketch of the data structure used to accumulate instructions for changing the flow network. It represents the demand path ABDC of Figure II-5 as the old flow path. Node C is shown as having had a RATATT dataset with two attributes in it. The changes illustrated are deletion of node B and substitution of node F for node D. The notation dA, etc., is used to represent the RLD delay coordinate for node A, etc.

The change pointers stack is the only kind built during flow network definition of modification that is not a pushdown stack. Instead it is a push<u>up</u> stack - new entries are added at the bottom rather than at the top. The reason is that the system dynamics node execution sequence normally

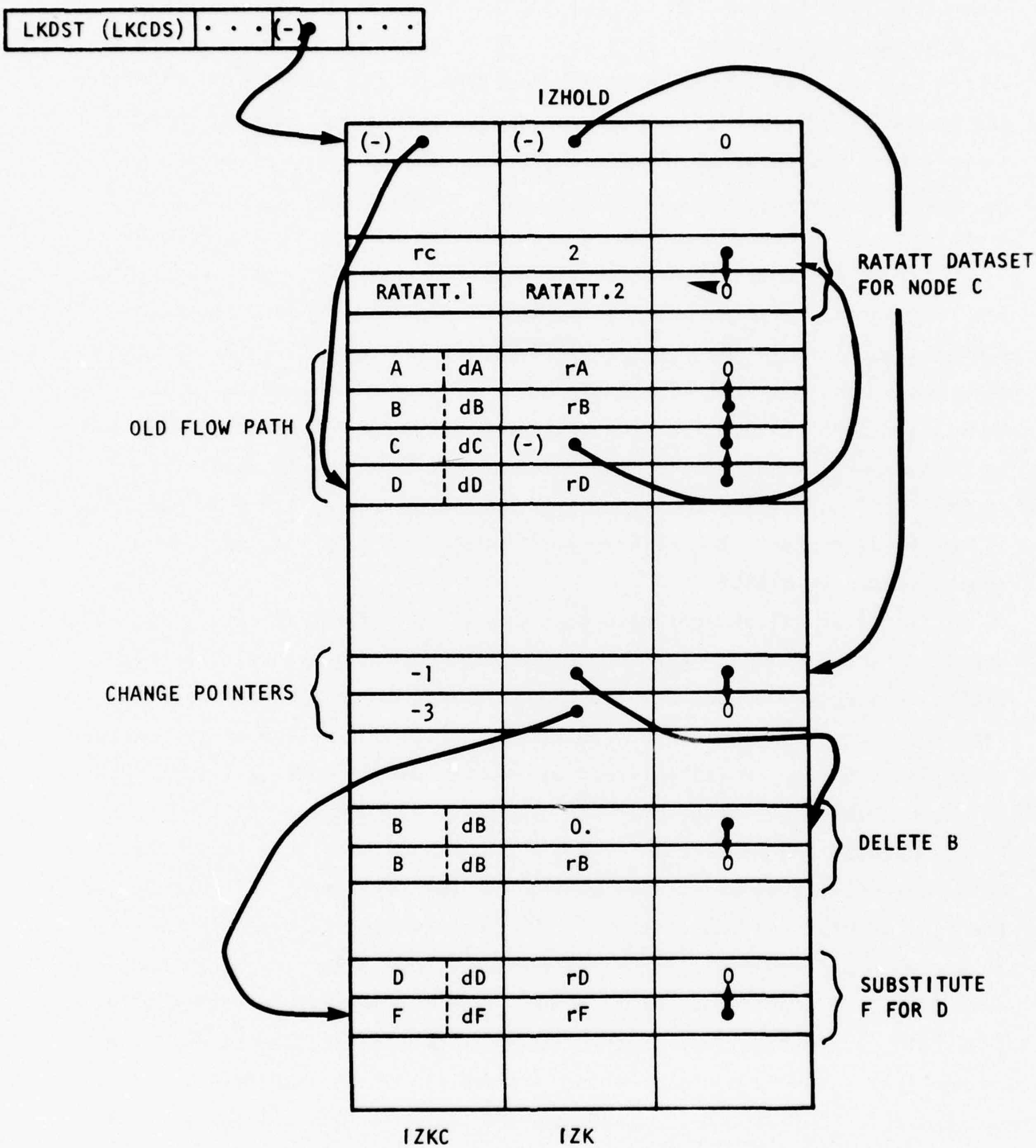


Figure III-5. Flow Path Being Changed

progresses from low echelon to high echelon, reflecting the flow of demand in an actual organization. This also facilitates the use of a single vRLDLK reference at a node in an intermediate or the top echelon to define all demand links leading to it from nodes it supports. Storing change notices in a pushup stack insures that, when they are processed top down by vRLFLK in the course of converting back to reference form, the implementation will proceed from low echelon to high echelon in the flow path.

At the completion of a setup pass in which flow network changes are introduced, all flow paths for which at least one change has been specified will be in the form shown in Figure III-5. All nodes directly affected by changes will have their LDSC-based membership data in the form shown in Figure III-3. The flow paths and membership linkages unaffected by changes will remain in reference form. For example, if dataset LKDST (LKCDS) in Figure III-5 contains a second flow path pointer and that path is not to be changed, the pointer will remain positive and the path will remain stored in DTABLE.

After all changes have been specified and all affected flow paths and membership linkages have been converted to the form just described, vRLFLK is executed to convert the changed parts of the network back to reference form. Unlike conversion after initial definition of the network, which is performed in its entirety by vRLFLK, vRLFLK must call on vMAPLK to execute the final stage of conversion when changes are involved. vRLFLK first converts the membership linkages from pushdown stacks to DTABLE and LCIX residency, just as it does during initial definition. It then processes the flow paths. Any that are newly defined are converted to DTABLE residency as during initial definition of the network. Any that are to be changed are only partially converted. First a duplicate of the stack containing the old flow path is created. vRLFLK then modifies this in accordance with the change instructions, and stores the revised flow path in DTABLE. But it retains the original old flow path and the intermediate pointer entry in IZHOLD. The resulting condition is shown in Figure III-6. The RATATT dataset associated with node C has been reestablished and its IZHOLD space released.

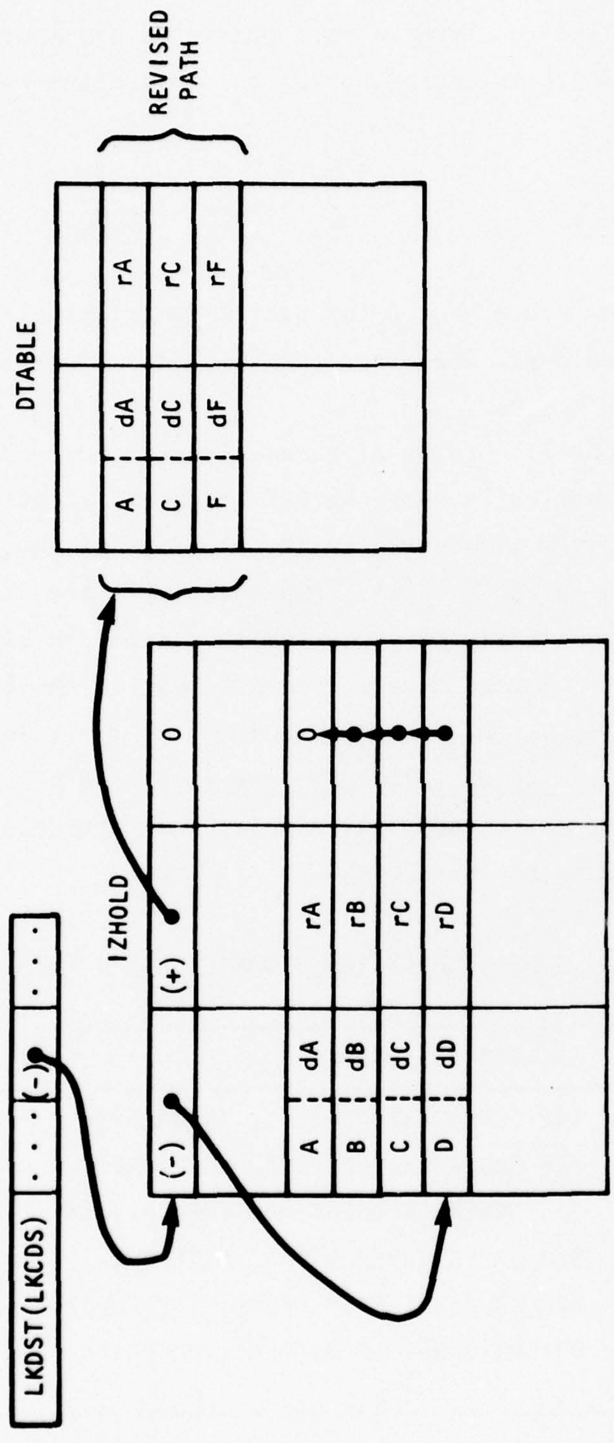


Figure III-6. Status of Changed Flow Path Entering MAPLK

After vMAPLK has performed any mapping required by NODMP datasets, it deletes the old path and the intermediate pointer entry from IZHOLD and stores the pointer to DTABLE in LKDST(LKCDS) to complete the conversion to reference form.

D. DELAYS DATA STRUCTURE

Two categories of data are stored for system dynamics delays. One is descriptors of generic delays. The other is the status of particular instances of the generic delays.

Array RLDNAM in /RLSYS/ contains alphanumeric descriptions of generic delays. Corresponding entries in array RLDPAR in /RLSYS/ contain two other descriptors. RLDPAR(1,*) is a code indicating the form of the delay. The codes used are shown in Table III-1. RLDPAR(2,*) is the mean delay if the delay form is exponential, or a pointer to a table in DTABLE if the delay form is boxcar. For boxcar delays, the entries in DTABLE consist of pairs (p_i, d_i) where p_i is the probability that the delay incurred by a unit of resource will be $\sum_{k=1}^{i} d_k$. That is, the entries are assumed to be stored in order of increasing cumulative delay, but the delay elements of the table are assumed to be incremental.

TABLE III-1. CODES FOR SYSTEM DYNAMICS DELAY FORMS

CODE	FORM
1	1st ORDER EXPONENTIAL, INFORMATION
2	1st ORDER EXPONENTIAL, MATERIAL
3	3rd ORDER EXPONENTIAL, INFORMATION
4	3rd ORDER EXPONENTIAL, MATERIAL
5	BOXCAR (ARBITRARY FORM), INFORMATION
6	BOXCAR (ARBITRARY FORM), MATERIAL

A generic delay is identified by the index of its parameters in RLDPAR (which is the same as the index of its description in RLDNAM).

When a generic delay is assigned for use in particular process or flow link, an RLD dataset is established. The normal sequence is that the model description and model input data assign the indexes of generic delays to data elements that associate a delay with a process. The same generic delay may be assigned to any number of distinct processes or flow link in the model. Then module ARLDIX is used to create an RLD dataset in which to simulate a particular instance of a delay. The coordinate assigned is the one plus the current value of DLRIXK. DLRIXK is initialized at zero. When the RLD coordinate is assigned, it is stored in place of generic delay index as the identifier of the delay associated with a process or flow link.

ARLDIX is called in two ways. It is called by RLDLY, the delay update module, if the delay identifier argument, DLRIX, is negative. That is, a negative identifier is assumed to be a generic delay index, and ARLDIX is called to replace it with an RLD coordinate. A positive DLRIX argument in RLDLY is assumed to be an RLD coordinate and dataset RLD(DLRIX) is updated. So generic delay identifiers specified as input data should be negative.

An exception to the requirement that generic delays be identified by negative indexes occurs in flow network definition. The delay arguments of NRLDLK, vCHDLK, etc., which do represent generic delays, are written as positive values in the model description, and vRLFLK employs ARLDIX directly (i.e., bypassing RLDLY) to assign RLD coordinates.

The content of an RLD dataset consists of the generic delay that it represents as the first element, followed by the intermediate rates and levels used to simulate system dynamics delays. The pattern for each delay form is shown in Table III-2. The element containing the output rate is also shown.

TABLE III-2. RLD DATASET LAYOUTS FOR SYSTEM DYNAMICS DELAYS

ELEMENT	RLID1	RLMD1	RLID3	DLMD3	RLIDB	RLMDB
1	INDEX OF GENERIC DELAY (FOR ALL FORMS)					
2	L	L	L1	L1	L1	L1
3		R	L2	L2	L2	R1
4			L3	L3	L3	L2
5				R1	L4	R2
6				R2	L5	L3
7				R3	L6	R3
.					.	.
.					.	.
.					.	.
OUTPUT RATE	L	R	L3	R3	L1	R1

ARRAY RL IN /RLSYS/ IS USED AS A BUFFER FOR PASSING THE CONTENT OF THE CURRENT RLD DATASET BETWEEN RLD AND ITS HELPERS RLID1, RLMD1, RLID3, ETC. THE CONVENTION USED IS THAT, IF RL(1) IS NOT POSITIVE OR ZERO, THEN THE DATASET HAS NOT BEEN STORED IN RL AND THE HELPER MODULE RLID1, ETC. MUST RETRIEVE IT.

E. STATISTICS COLLECTION DATA STRUCTURE

The statistics collection system implemented in MAWLOGS for system dynamics variables assumes that all variables to be observed are stored in PDS datasets (or, potentially, are pointed to from PDS datasets, as is true of flow rates in the flow path network, for example). Variables are flagged for observation on input via vRLPDS. Only variables belonging to datasets identified in array RLDST in /RLSYS/ are eligible for flagging. The observation of a variable is characterized by a form in which the statistics are to be recorded and a frequency of observation. The available forms are shown in Table III-3. Nine frequencies are provided for. Frequency code i, i=1,2,...,9 means an interval between observations of (2^{i-1}) RLSTIU(1), where RLSTIU(1) is the basic collection interval input to vRLPDS, rounded to the nearest integral number of time steps. RLSTIU is required to be at least one time step. From RLSTIU(1) rounded to the nearest integral number of time steps, arrays RLSTIU and RLSTIS are constructed. RLSTIU(i) contains the collection interval corresponding to frequency code i, in simulation time units. RLSTIS(i) expresses RLSTIU(i) in time steps. Both arrays are in /RLSYS/.

Each variable to be observed is represented by an entry in arrays KRLST and KRLSTI in /RLSYS/. Variables are assigned to the arrays in the order input by vRLPDS. The entries for the ith variable contain:

KRLST(i)			KRLSTI(i)	
freq code	type code	i	STATI index	index in DSPOOL of variable

The content shown for KRLST is only the initial form assigned by RLSASN during vRLPDS input. At the same time that KRLST and KRLSTI are set as shown, additional descriptors are stored in RLSNAM in /STNAM/ as follows (for the ith variable):

TABLE III-3. FORMS OF STATISTICS COLLECTION

CODE	COLLECTING ROUTINE	DESCRIPTION
0		DISCONTINUE OBSERVATION
1	PRRLST	PRINT CURRENT VALUE
2		NOT USED; DEFAULTS TO PRRLST
3	COLCT	TIME-WEIGHTED MEAN, STANDARD DEVIATION, AND RANGE
4	HISTO	COLCT PLUS FREQUENCY DISTRIBUTION
5	TACTN	RECORD ON MAWLOGS TRANSACTION FILE CURRENT VALUE AND TIME SINCE LAST OBSERVATION

RLSNAM(1,i)		RLSNAM(2,i)	RLSNAM(3,i)	RLSNAM(4,i)
RLDST index of var's dataset type	index of var in dataset	1st coord of dataset	2nd coord of dataset	3rd coord of dataset

Finally, if the collection form is one of the STATI forms COLCT, HISTO, or TACTN, STATI identifiers are generated and stored in the appropriate array in /STNAM/ using module IXSTI. The identifiers, generated by RLSASN, are

INOD	reserved for PERMAT variables	XIDST			XIDRES	
system dynamics node number	not used	RLDST index	element index in dataset	19	resource number	priority or zero

The STATI and IXSTI nomenclature associated with these identifiers is shown above the rectangles. These data are stored in the STATI identifier arrays in the order they are assigned as they are read in via vRLPDS.

The connections among the variable and its identifiers are summarized in Figure III-7. It has been assumed, for illustration, that the variable is to be subjected to COLCT observation. Dotted lines show the relations that maintain the logical identity of the variable as an element of a PDS dataset. During statistics collection, KRLST defines the type of collection and KRLSTI makes it possible to observe the value directly in its DSPOOL storage location.

After all variables have been flagged for observation, module RLSSET is executed to reorganize KRLST for greater efficiency. First it is sorted into ascending order, to group the variables by collection frequency, with the highest frequencies at the top of the array. Then the array LRLST(i), i=1,2,...,9 is set. LRLST(i) is the index in (sorted) KRLST of the last entry requiring the ith collection frequency. After LRLST is set, the frequency code is stripped off the KRLST entries. Note that the relation between a KRLST and its corresponding KRLSTI has been preserved by the index in the right side of the KRLST.

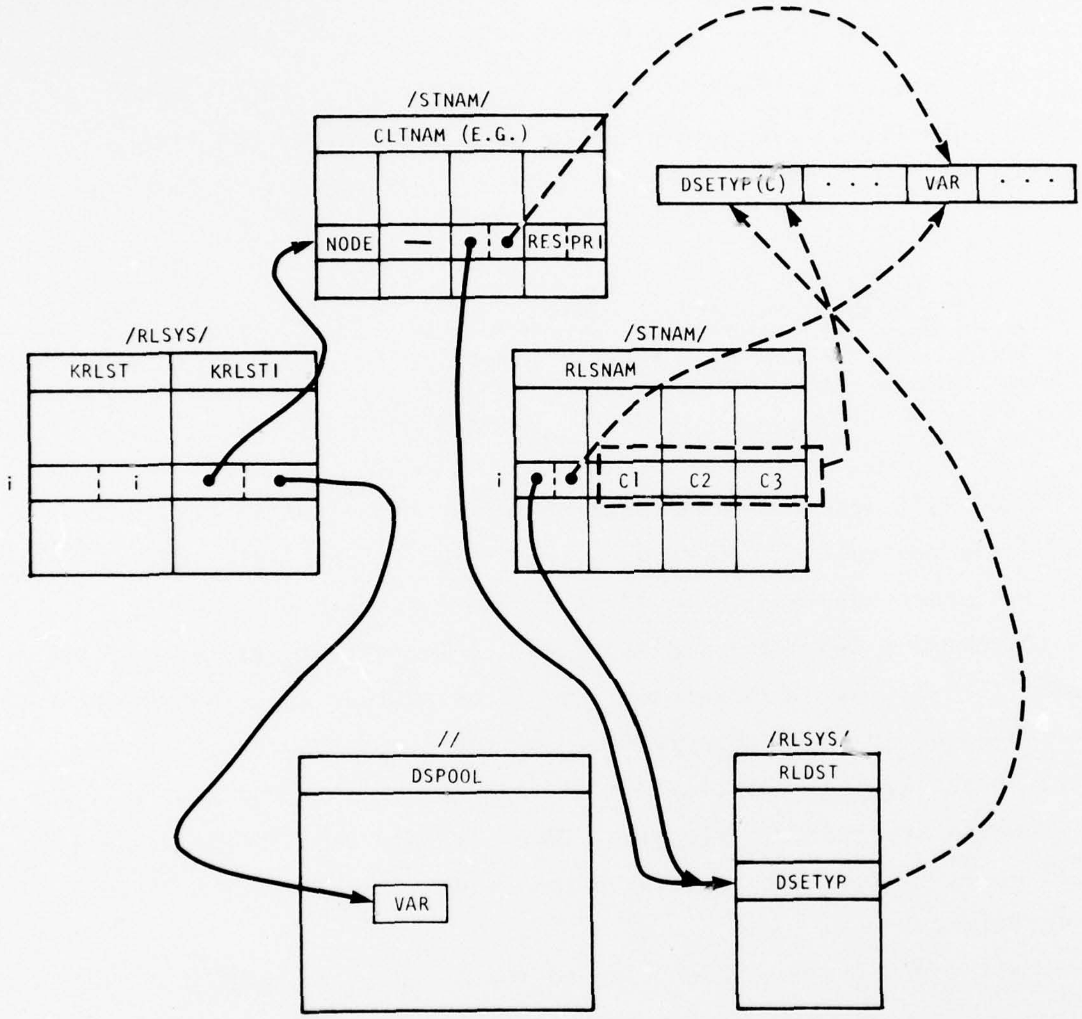


Figure III-7. Relations Among Statistics Descriptors

When changes are introduced to the statistics collection pattern, via vCHPDS inputs, the frequency codes are reestablished in KRLST. After the changes have been read in, KRLST is resorted so that any changes to collection frequency can be properly reflected and LRLST is reset. The frequency code is then stripped off again.

As the simulation progresses, the next collection time for the ith frequency is maintained in TNXT(i) in /RLSYS/.

TABLE III-4. LABELED COMMON /RLSYS/

VARIABLE	DESCRIPTION	INITIALIZING MODULE
	MODEL CONTROL	
TSTEP	TIME STEP, IN NUMBER OF SIMULATION TIME UNITS	DATAN & vIZRL
NSTEP	NUMBER OF TIME STEPS EXECUTED TO DATE	BLKRL & vIZRL
NSTHH	N STEP THRESHHOLD (OBSOLETE; NOT USED)	NA
NSTW	NUMBER OF TIME STEPS TO BE USED FOR ACCELERATED WARMUP OF FLOW RATES SUBJECT TO DELAY	DATAN & vIZRL
RLNOD	CURRENT RL NODE NUMBER (FLOATING POINT)	NA
KRLNOD	CURRENT RL NODE NUMBER (INTEGER)	NA
RLRES	RL RESOURCE NUMBER (e.g., FLOW CLASS NUMBER)	NA
RLSW	CONTROL VARIABLE MAINTAINED BY vRLCTR - EQUALS LEVELS WHEN LEVELS ARE TO BE UPDATED, RATES WHEN RATES ARE TO BE UPDATED, AND NEXT WHEN CONTROL IS TO BE GIVEN TO ACTIVITIES IN WHICH NEITHER RATES NOR LEVELS ARE TO BE UPDATED	BLKRL & vIZRL
LEVELS	VALUE USED BY RLSW TO SIGNAL LEVELS UPDATE	BLKRL
RATES	VALUE USED BY RLSW TO SIGNAL RATES UPDATE	BLKRL
RLMODE	CONTROL VARIABLE MAINTAINED BY vRLCTR AND vRLMSU - EQUALS SIM WHEN SIMULATION LOGIC IS TO BE EXECUTED; EQUALS SETUPM WHEN LINK DEFINITION OR MODIFICATION LOGIC IS TO BE EXECUTED	BLKRL & vIZRL
SETUPM	VALUE USED BY RLMODE TO SIGNAL SET-UP MODE	BLKRL
SIM	VALUE USED BY RLMODE TO SIGNAL SIMULATION MODE	BLKRL
RLNNAM(I)	NAME OF Ith SYSTEM DYNAMICS NODE (I=1,2,... NRLN)	vRLNOD
NRLN	NUMBER OF SYSTEM DYNAMICS NODES IN CURRENT MODEL	BLKRL
NRLNMX	MAXIMUM NUMBER OF SYSTEM DYNAMICS NODES CURRENT MODEL CAN ACCOMMODATE (DIMENSION OF RLNNAM)	cRLSIZ*

TABLE III-4. LABELED COMMON /RLSYS/ (CONTINUED)

VARIABLE	DESCRIPTION	INITIALIZING MODULE
	LINKAGE DEFINITION AND ACCESS	
LKDST	NAME OF A PDS DATASET TYPE USED TO IDENTIFY FLOW PATHS OF CURRENT INTEREST	NA
LKCDS(I)	Ith COORDINATE OF PDS DATASET OF TYPE LKDST WHOSE FLOW PATHS ARE OF CURRENT INTEREST ($I \leq 5$)	NA
IXDRAT	INDEX OF ELEMENT IN LKDST(LKCDS) CORRESPONDING TO DEMAND FLOW PATHS OF CURRENT INTEREST	NA
IXRRAT	INDEX OF ELEMENT IN LKDST(LKCDS) CORRESPONDING TO RETURN FLOW PATH OF CURRENT INTEREST	NA
DRATE	DEMAND FLOW RATE ASSOCIATED WITH DEMAND LINK OF CURRENT INTEREST	NA
RRATE	RETURN FLOW RATE ASSOCIATED WITH RETURN LINK OF CURRENT INTEREST	NA
RATDST	ARRAY OF NAMES OF DATASET TYPES IN THIS MODEL THAT OWN FLOW PATHS	RLLNK
	DELAYS	
RL(I)	Ith ELEMENT OF SYSTEM DYNAMICS DELAY CURRENTLY BEING DEFINED OR UPDATED (I=1,2,...,20)	NA
RLDPAR(*,J)	PARAMETERS FOR Jth SYSTEM DYNAMICS DELAY, J=1,2,...,NRLDPR, AS FOLLOWS:	
	RLDPAR(1,J) - CODE INDICATING FORM OF DELAY	vIRLD
	RLDPAR(2,J) - MEAN DELAY TIME FOR 1st AND 3rd ORDER DELAY; POINTER TO DTABLE FOR BOXCAR DELAYS	
RLDNAM(*,J)	ALPHANUMERIC DESCRIPTION OF Jth DELAY (J=1,2,...,NRLDPR)	vIRLD
NRLDPR	MAXIMUM NUMBER OF SYSTEM DYNAMICS DELAY TYPES PERMITTED IN CURRENT MODEL (DIMENSION OF RLDPAR, RLDNAM)	cRLSIZ*
DLRIXK	LAST INDEX ASSIGNED AS THE COORDINATE OF AN RL DELAY	BLKRL
RLDTYP(I)	SHORT NAME OF DELAY FORM OF CODE I, I=1,2,...,6 (SEE TABLE III-1)	BLKRL

TABLE III-4. LABELED COMMON /RLSYS/ (CONTINUED)

VARIABLE	DESCRIPTION	INITIALIZING MODULE
	STATISTICS COLLECTION	
RLDST(I)	Ith ENTRY IN LIST OF PDS DATASET TYPE NAMES CONTAINING ELEMENTS POTENTIALLY SUBJECT TO SYSTEM DYNAMICS STATISTICS COLLECTION (I=1,2,...,NRLDST)	BLKRL
KRLST(I)	PARTIAL IDENTIFIER (SEE KRLSTI) OF Kth SYSTEM DYNAMICS STATISTICAL VARIABLE - CONTAINS (RLSTAT COLLECTION CODE *10000) + K, K=1,2,...,NXRLST	RLSASN & vIZRL
KRLSTI(K)	REMAINDER (AFTER KRLST(I)) OF IDENTIFICATION OF Kth SYSTEM DYNAMICS STATISTICAL VARIABLE - CONTAINS (STAT1 INDEX *100000) + INDEX IN DSPOOL OF VALUE TO BE OBSERVED	RLSASN & vIZRL
NRLST	MAXIMUM NUMBER OF SYSTEM DYNAMICS VARIABLES ON WHICH STATISTICS CAN BE COLLECTED (SIZE OF ARRAYS KRLST AND KRLSTI, AND OF RLSNAM IN /STNAM/)	cRLSIZ*
NXRLST	LAST INDEX IN KRLST, ETC. ASSIGNED	BLKRL (cRLSIZ)
TNXT(I)	NEXT SIMULATION TIME AT WHICH SYSTEM DYNAMICS STATISTICS WHOSE INTERVAL BETWEEN OBSERVATIONS IS ((2**(I-1))* BASIC COLLECTION INTERVAL) ARE TO BE OBSERVED (I=1,2,...,9)	vRLPDS & vIZRL
LRLST(I)	INDEX IN KRLST, ETC. OF LAST SYSTEM DYNAMICS STATISTIC TO BE OBSERVED AT THE Ith FREQUENCY	RLSSET
NKRLST	(NOT USED)	
NRLDST	NUMBER OF RATE/LEVEL DATA SET TYPES	
NRDTMX	DIMENSION OF ARRAY RLDST	cRLSIZ*
IRLST	INDEX IN KRLST OF STATISTIC CURRENTLY BEING COMMANDED	NA
PRLSSW	PAGE HEADER CONTROL SWITCH FOR USE IN PRINTING SYSTEM DYNAMICS STATISTICS	NA
COLTYP	COLLECTION TYPE CODE, AS FOLLOWS: 1 - PRINT CURRENT VALUE 2 - NOT USED (DEFAULT IS SAME AS 1) 3 - COLCT 4 - HISTO 5 - TACTN	
IXDSPL	INDEX IN DSPOOL OF VALUE TO BE OBSERVED FOR CURRENT STATISTIC	NA

TABLE III-4. LABELED COMMON /RLSYS/ (CONTINUED)

VARIABLE	DESCRIPTION	INITIALIZING MODULE
	STATISTICS COLLECTION (CONTINUED)	
RLSTIU(I)	Ith STATISTICS COLLECTION INTERVAL, IN SIMULATION TIME UNITS (I=1,2,...,9)	vRLPDS
RLSTIS(I)	Ith STATISTICS COLLECTION INTERVAL, IN TIME STEPS	vRLPDS
MRLST	INDEX OF ENTRY IN KRLSTI CORRESPONDING TO KRLST(IRLST)	NA

*cRLSIZ REFERS TO THE "COMMON" MODULE, AN INPUT TO THE MODEL ASSEMBLER. THE MODULE CONTAINS FORTRAN DATA STATEMENTS INTO WHICH VALUES READ BY THE ASSEMBLER IN THE DIMENSIONS MODULE ARE INSERTED AT ASSEMBLY TIME. MODULE RLSIZ IS THEN INCORPORATED IN BLKRL BY MEANS OF A *CALL RLSIZ STATEMENT THERE.

TABLE III-5. LABELED COMMON /LC/

VARIABLE	DESCRIPTION	INITIALIZATION MODULE
RLC(*,I)	Ith UNIQUE SET OF PDS COORDINATES FOR DATASETS THAT REPRESENT OR OWN SYSTEM DYNAMICS FLOW PATHS	RLLNK
LCIX(*)	ARRAY OF POINTERS INTO RLC	vRLFLK
NRLDIM	MAXIMUM NUMBER OF COORDINATES PERMITTED IN AN ENTRY IN RLC (1st DIMENSION OF RLC)	cRLSIZ*
NRLC	MAXIMUM NUMBER OF ENTRIES PERMITTED IN RLC (2nd DIMENSION OF RLC)	cRLSIZ*
NCLIX	DIMENSION OF LCIX	cRLSIZ*
NXTRLC	INDEX OF NEXT AVAILABLE POSITION IN RLC	BLKRL
NXTLCX	INDEX OF NEXT AVAILABLE POSITION IN LCIX	BLKRL
NFRELC	NUMBERS OF POSITIONS IN LCIX THAT HAVE BEEN FREED FOR REUSE	BLKRL
LRLCHG	USED BY SAVLK TO SIGNAL (=1) LRLOOP THAT AN LCIX LIST HAS BEEN TRANSFERRED TO IZHOLD	
LRLPTR	INDEX IN IZHOLD OF TOP OF LCIX STACK THAT WAS TRANSFERRED (AND SIGNALLED BY LRLCHG) SET TO -1 BY LRLOOP WHEN NO LRLOOP IN PROGRESS WHEN LOOP IS IN PROGRESS, IT IS INDEX IN LCIX OR IZHOLD OF CURRENT STACK ELEMENT USED BY LRLOOP 1 - STACKS LOOP 0 - LCIX LOOP	

CHAPTER IV

CONTINUOUS SERVICE VERBS

THIS PAGE LEFT INTENTIONALLY BLANK

ADDLK (NODOLD, NODNEW, DLYNEW, IREL)

General Description

Inserts RLNODE named NODNEW into demand flow path pointed to from LKDST (LKCDS). IXDRAT. RL delay type with number DLYNEW is assigned to the new link leading into NODNEW. NODOLD is the name of the RLNODE before or after which NODNEW is to be inserted. IREL indicates before or after as follows -

IREL	NODNEW Relation to NODOLO
-1	BEFORE
1	AFTER

Assembler Inputs

Arguments.

NODOLD - name of RL node before or after which NODNEW is to be inserted

NODNEW - name of RL node to be inserted into demand flow path

DLYNEW - delay type assigned to new link leading into NODNEW

IREL - indicates relation between NODNEW and NODOLD

Parameter Slots. None.

Examples

Verb ADDLK is used to add a demand link during a model run, generally to an additional source node.

```
CPASI (P = 0, 1  $
       1 = BRLKD (P = 2  $
                  2 = STGBR (P = 79  $
                             2 = ADDLK (P = * CONUS, * WRSL, 0.,-1))))
       + ADD DEMAND LINK TO WRSL BEFORE CONUS LINK
       + IN D2 DEMAND PATH
```

GASP Files Used. None.

Permanent Attributes Accessed

(ADDLK-1)

In /RLSYS/:

IXDRAT - index of element in LKDST (LKCDS) corresponding to demand flow paths of current interest

DRATE - demand flow rate associated with demand link of current interest

Verb Inputs

From Calling Program.

Arguments - see above

From RLLNK. None.

Verb Outputs

To RLLNK (1st call)

Arguments:

NODNEW - name of RL node to be inserted into demand flow path

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

DRATE - demand flow rate associated with demand link of interest

DLYNEW - number of rate level delay type associated with the link

To RLLNK (2nd call)

Arguments:

NODOLD - name of RL node before or after which NODNEW is to be inserted

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

-2 - insert link after an existing one

0 - indicate there is to be no delay

To RLLNK (3rd call)

NODOLD - name of RL node before or after which NODNEW is to be inserted

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

-4 - insert link before an existing one

0 - indicates there is to be no delay

(ADDLK-2)

Programs Called

Verbs. None.

Other. RLLNK.

Input/Output Files Used.

NPRNT - Print File

(ADDLK-3)

THIS PAGE LEFT INTENTIONALLY BLANK

ADRLK (NODOLD, NODNEW, DLYNEW)

General Description

Inserts return flow link to RLNODE named NODNEW after existing link to RLNODE named NODOLD in rate stack pointed to from LKDST (LKCDS). /// IXRRAT. RL delay type DLYNEW is assigned the new link.

Assembler Inputs

Arguments.

NODOLD - old node after which new link is to be inserted in rate stack

NODNEW - new node to be added to existing return path

DLYNEW - type of delay assigned to new link

Parameter Slots. None.

Examples

Verb ADRLK is used to add a receipt link during a model run, from a new source node.

```
ACSUP (2 = STGBR (P = 28  $
                  1 = RLRLK (P = * NODEB, 3.)  $ + R1 PATH
                  2 = ADRLK (P = * NODEA, * NODEB, 4.)))
                  + ADD LINK TO R1 PATH
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

IXRRAT - index of element in LKDST (LKCDS) corresponding to return flow paths of current interest

RRATE - return flow rate associated with return link of current interest

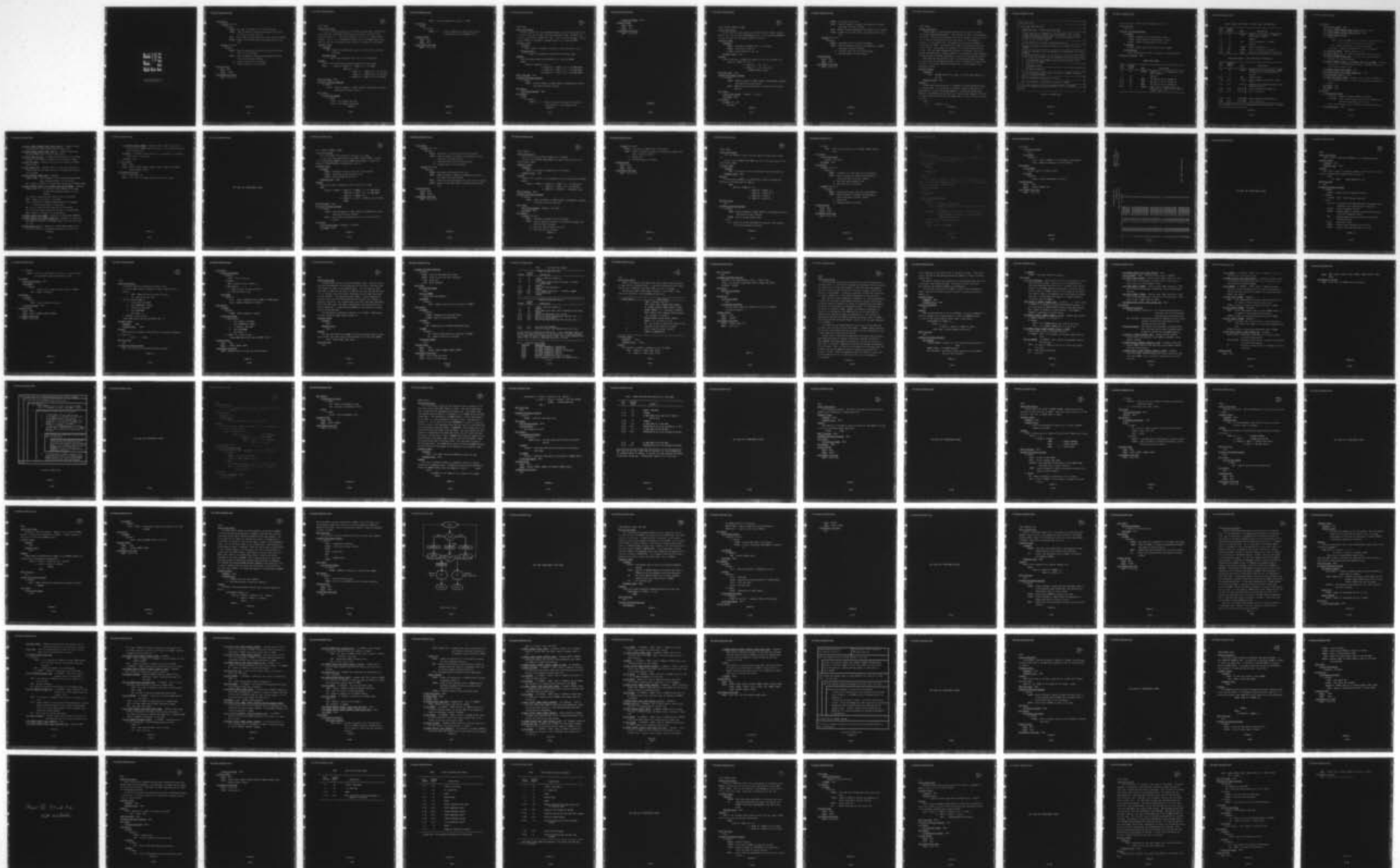
Verb Inputs

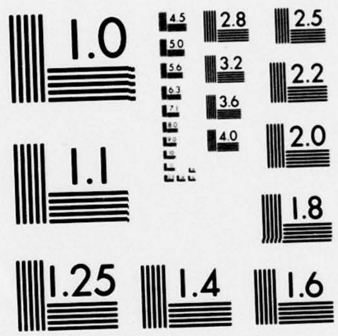
From Calling Program. Arguments - see above

From RLLNK. None.

AD-A040 804 MODELS OF THE US ARMY WORLDWIDE LOGISTIC SYSTEM (MAWLOGS) VOLUME III MODU..(U) BDM CORP VIENNA VA FEB 77 BDM/W-76-211-TR-VOL-3-PT-6 DAAG39-76-C-0134 2/4

UNCLASSIFIED F/G 15/5 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Verb Outputs

To RLLNK (1st call)

Arguments:

NONNEW - new node to be added to existing return path

IXRRAT - index of element in dataset corresponding to return flow path of current interest

RRATE - return flow rate associated with return link of interest

DLYNEW - number of rate level delay type associated with the link

To RLLNK (2nd call)

Arguments:

NONOLD - name of old node with existing link after which new link is to be inserted

IXRRAT - index of element in dataset corresponding to return flow path of current interest

-2 - insert link after an existing one

0 - indicate there is to be no delay

Programs Called

Verbs. None.

Otherr. RLLNK.

Input/Output Files Used.

NPRNT - Print File.

(ADRLK-2)

BRLKD (NBR)

General Description

For use in defining branches in a rate/level demand path. NBR branches may be defined. The number associated with a branch is that of the PS executed to define it. The pointer word for a branch is in LKDST (LKCDS). (IXDRAT + IBR-1). It is assumed that LKDST, LKCDS, and IXDRAT have already been set, and that the demand pointer words are contiguous in the dataset.

Assembler Input

Arguments.

NBR - number of branches which may be defined in this rate level flow path

Parameter Slots.

1 - verb to be executed for each link path to be defined

Examples

BRLKD is used to define more than one demand path for a node.

```
NODEA. . . . ACSUP (2 = RLRLK (P = * NODEA, 3.), + R1 CONUS
                        BRLKD (P = 2  $
                                  1 = RLDLK (P = * NODEA, 0.), + D1 TO NODEA
                                      RLDLK (P = * CONUS, 0.)$ + D1 TO CONUS
                                  2 = RLDLK (P = * WRSL, 4.))) + D2 TO WRSL
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

IXDRAT - index of elements in LKDST (LKCDS) corresponding to demand flow path of current interest

Verb Inputs

From Calling Program.

In /RLSYS/:

LKDST - link dataset type name

LKCDS - link dataset coordinates

(BRLKD-1)

IXDRAT - index to demand rate pointers in LKDST

Verb Outputs

To PSn.

In /RLSYS/:

IXDRAT + n - 1 - index of demand rate stack to be used for links defined in this parameter slot

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used.

NPRNT - Print File

(BRLKD-2)

BRLKR (NBR)

General Description

Splits a return rate flow into NBR branches by permitting the definition of distinct flow links in each of up to 20 parameter slots. Variable IXR in RLSYS is modified to become IXR + IBR - 1 when PS IBR is executed. Assumes multiple RPTRS are contiguous in dataset beginning at incoming IXRRAT.

Assembler Inputs

Arguments.

NBR - number of branches into which a return rate flow is split

Parameter Slots.

1 - verbs to be executed for each branch to define links

Examples

Multiple receipt paths can be defined for a node with BRLKR.

```
NODEA.
    ACSUP (2 = BRLKR (P = 3  $
                  1 = RLRLK (P = * NODEA, 3.)  $ + R1 FROM NODEB
                  2 = RLRLK (P = * NODEA, 5.)  $ + R2 FROM NODEC
                  3 = RLRLK (P = * NODEA, 6.)))  + R3 FROM NODED
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

IXRRAT - index of element in LKDST (LKCDS) corresponding to return flow path of current interest

Verb Inputs

From Calling Program. None.

Verb Outputs

To PSn.

In /RLSYS/:

IXRRAT + n - 1 - index of receipt rate stack to be used for links defined in this parameter slot

(BRLKR-1)

To Calling Program. None.

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

NPRNT - Print File

(BRLKR-2)

CHDLK
S/SS/CS
4/75

CHDLK (NODOLD, NODNEW, DLYNEW)

General Description

In the rate flow stack pointed to by LKDST (LKCDS), IXDRAT, replaces demand link to the RL node named NODOLD by one to the node named NODNEW and assigns the delay with parameters RLDPAR (*, DLYNEW).

Assembler Inputs

Arguments.

NODOLD - terminator of demand link to be replaced

NODNED - terminator of new link

DLYNEW - delay assigned to new link

Parameter Slots. None.

Examples

CHDLK can be used to change the supporting node during a model run.

```
ACSUP (2 = STGBR (P = 16  $
                  1 = RLDLK (P = * 1GS, 3.)  $
                  2 = CHDLK (P = * 1GS, *4GS, 0.)))
                     + CHANGE SUPPORTING GSUS
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

IXDRAT - index of element in LKDST (LKCDS) corresponding to demand flow paths of current interest

DRATE - demand flow rate associated with demand link of current interest

Verb Inputs

From Calling Program. Arguments - see above

From RLLNK. None.

Verb Outputs

To RLLNK (1st call)

Arguments:

NODNEW - terminator of new link

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

DRATE - demand flow rate associated with demand link of interest

DLYNEW - number of rate level delay type associated with the link

To RLLNK (2nd call)

Arguments:

NODOLD - terminator of old link to be replaced

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

-3 - substitute new link for an existing one

0 - indicate there is to be no delay

Programs Called

Verbs. None.

Other. RLLNK.

Input/Output Files Used

NPRNT - Print File

(CHDLK-2)

CHRLD (INFIL)

General Description

Reads changes to system dynamics delay parameters from file INFIL, and revises affected RLD datasets. Input formats are shown in the accompanying tables. Both form and the delay times may be changed. The only restriction is that a revised form be of the same category - information or material - as the current form. Any number of delays may be changed in an execution of CHRLD. The activities of CHRLD are outlined in the accompanying figure. Each change is processed in its entirety before the next is read. The individual changes are printed as encountered. After all changes have been made, a complete report of system dynamics delay parameters and descriptors is printed, and control is returned to the calling routine.

Note that the time width of a cell in an exponential delay is deemed to be 2.3 times its average delay width. That corresponds to a range about the mean of a 1st order delay that extends from 0 to 90 percent of the probability density.

Assembler Inputs

Arguments.

INFIL - FORTRAN logical file number from which CHRLD data are to be read

Parameter Slots. None.

Examples

vCHRLD would normally be one in a sequence of change modules executed at selected times in a simulation to introduce changes to the system. A good practice is to put the change modules in a subnode of nZINIT whose execution is scheduled via the time file. Assuming that subnode 3 of nZINIT is used and that the input file for vCHRLD data is logical file 5, the setup would be

```
ZINIT. . . .
  /3/ . . . , CHRLD (P = 5), . . .
```

(CHRLD-1)

- Read header card
- Loop through remaining cards
  - Read next card. If end of data, end loop.
  - Insure that card is a CHRLD card, that the delay index is within range, and that the new delay type is the same as it is currently relative to information and delay. Print error messages if not proper in any of these respects.
  - If new delay form is boxcar, read table of points.
  - Construct cumulative times for segments of old delay, for use by RLDREV
  - Construct cumulative times for segments of revised delay, for use by RLDREV
  - Loop through RLD datasets that use parameters of current delay index
    - For each RLD, call RLDREV to revise rates and levels of the cells of the delay
    - Release old RLD
    - Create revised RLD
  - If revised delay is boxcar type, store table in DTABLE, releasing old if it was also boxcar
  - Print message describing current revision
  - Set new parameters into RLDPAR, new name into RLDNAM
- If there were changes, print delay parameters report with RLDRPT
- RETLOG

Outline of vCHRLD Logic

(CHRLD-2)

Execution of this subnode could be schedule via vFILE.

GASP Files Used

Time File.

Permanent Attributes Accessed

In /RLSYS/.

RLDPAR - parameters of system dynamics delays

RLDNAM - nomenclatures of system dynamics delays

RLDTYP - names of system dynamic delay forms

In /TBLCOM/.

DTABLE - boxcar delay distributions to be changed

In PDS Datasets.

RLD - RLD datasets of forms or parameters that were changed

Statistics Collected. None.

HEADER CARD FORMAT

CARD COLS	FORTRAN FORMAT	NAME	DESCRIPTION
1-6	A6	NAMCRD	"CHRLD1", or, if no changes are to be read, "ENDRLD"
7	1X		ignored
8-13	A6	NAM1	1st word of name of change set
14-19	A6	NAM2	2nd word of name of change set
20-25	A6	NAM3	3rd word of name of change set
26-30	I5	NONFAT	fatal (>0) or nonfatal (≤0) error indicator-run is aborted via ZERRPT if "fatal" flag is on

(CHRLD-3)

CHRLD1 CHANGE CARD FORMAT - FOR FORM, NAME, AND MEAN DELAY

CARD COLS	FORTRAN FORMAT	NAME	DESCRIPTION
1-6	A6	NAMCRD	"CHRLD1" for a change, or "ENDRLD" to signal end of changes
7	1X		ignored
8-10	I3	IXD	index of parameters in RLDPAR
11	1X		ignored
12-16	A5	TYPNAM	name of revised form
17-26	F10.3	DPARAM	mean if new delay is exponential
27-80	9A6	RLDNM	descriptive nomenclature of new delay

CHRLD2 CARD FORMAT - FOR BOXCAR DELAY DISTRIBUTION

CARD COLS	FORTRAN FORMAT	NAME	DESCRIPTION
1-6	A6	NAMCRD	"CHRLD2" for distribution data, "ENDDT" to signal end of distribution data
7	1X		ignored
8-10	I3	IXD1	index of delay
11-16	F6.0	DT(2,k)	delay increment for kth cell, k=1,8, or 15
17-20	F4.4	DT(1,k)	probability of cumulative delay corresponding to kth cell
21-26	F6.0	DT(2,k+1)	delay increment and probability
27-30	F4.4	DT(1,k+1)	for k + 1st cell*
.			
.			
.			
71-76	F6.0	DT(2,k+6)	delay increment and probability
77-80	F4.4	DT(1,k+6)	for k + 6th cell*

*When a probability entry ≤0 is encountered, the previous entry is treated as the last in the distribution.

(CHRLD-4)

Verb Inputs

From Calling Program. None.

From (both) INEROR (NAMCRD, BLANK, CHRLD, ONE, 0). None.

From INEROR (NAMCRD, BLANK, CHRLD, TWO, 0). None.

From IDSLON (IXDO, RLD, NE).

Arguments -

IXDO - index in DSPOOL of first element of current RLD dataset, or ≤0 if all RLDs have been accessed

NE - number of elements in current RLD dataset

In /DPTRS2/ - COORD(1), coordinate of current RLD dataset

From CLEAR (XN, NNEW). Argument XN set to zero.

From CLEAR (RL(2), 19). All except 1st element of RL set to zero.

From DLYLEV (DSPOOL (IXDO), NEL). Function value, XLEV, amount of resource in current RLD dataset

From RLDREV (DSPOOL (IXDO + 1), TO, NOLD, INCO, XN, TN, NNEW). Argument XN, resource in current RLD redistributed in pattern reflecting revised form and delay times

From PERMDS (RELSET, RLD, COORD). None.

From PERMDS (ADDSET, RLD, COORD, NELNEW, RL). None.

From RLDTB (NOLD, IDT1-1). None.

From INDTB (DT, NNEW, IXDTAB). Argument IXDTAB, index in DTABLE of entry immediately prior to first of new boxcar distribution just stored

From RLDRPT. None.

From ZERRPT. None.

Verb Outputs

To Permanent Attributes.

In /RLSYS/ - changes to RLDPAR, RLDNAM, and RLDTYP

In /TBLCOM/ - space released when old boxcar was replaced, and new space occupied by new boxcars

In PDS - RLD datasets revised, released, and added as required

To Calling Program. None.

(CHRLD-5)

To (both) INEROR (NAMCRD, BLANK, CHRLD, ONE, 0). Arguments identifying would-be CHRLD1 card as locus of input error

To INEROR (NAMCRD, BLANK, CHRLD, TWO, 0). Arguments identifying would-be CHRLD2 card as locus of input error

To IDSLON (IXDO, RLD, NE). Argument RLD, dataset type to be looped through. Also IPDRET in /PDSRET/, return address in CHRLD.

To CLEAR (XN, NNEW). Arguments XN, array to be cleared, and NNEW, length of XN.

To CLEAR (RL(2), 19). Arguments RL(2), 1st entry of array to be cleared, and 19, the number entries in RL (beginning at RL(2)) to be cleared

To DLYLEV (DSPOOL (IXDO), NEL). Arguments -

DSPOOL (IXDO) - first element of array containing delay dataset whose resource content is to be derived

NEL - number of elements in delay dataset beginning at DSPOOL (IXDO)

To RLDREV (DSPOOL (IXDO + 1), TO, NOLD, INCO, XN, TN, NNEW). Arguments -

DSPOOL (IXDO + 1) - first rate or level element of old delay dataset whose resource content is to be revised

TO - array of cumulative end times of cells in old delay

NOLD - number of time cells in old delay

INCO - increment for indexing through levels of old dataset (2 for material boxcar, 1 otherwise)

XN - array in which revised levels are to be returned

TN - array of cumulative end times of cells in revised delay

NNEW - number of time cells in revised delay

To PERMDS (RELSET, RLD, COORD). Arguments to release RLD (COORD(1))

To PERMDS (ADDSET, RLD, COORD, NELNEW, RL). Arguments to create new RLD (COORD(1)) dataset with NELNEW elements and rates and levels as in RL.

To RLDTB (NOLD, IDT1-1). Arguments to release NOLD elements (old boxcar distribution) from DTABLE beginning at position IDT1.

(CHRLD-6)

To INDTB (DT, NNEW, IXDTAB). Arguments NNEW, number of new entries to be stored in DTABLE, and DT, elements to be stored (new boxcar delay distribution)

To RLDRPT. In /RLSYS/, delay descriptors and parameters; in /TBLCOM/, boxcar delay distributions

To ZERRPT. None.

Programs Called

Verbs. None.

Other. ADDSET, CLEAR, DLYLEV, IDSLON, INDTB, INEROR, LIN, PERMDS, RELSET, RLDREV, RLDTB.

Input/Output Files Used

INFIL - CHRLD input file

NPRNT - print file, for change report and parameters report

THIS PAGE LEFT INTENTIONALLY BLANK

CHRLK (NODOLD, NODNEW, DLYNEW)

General Description

This verb changes the receipt link structure for a node.

In the return flow stack pointed to by LKDST (LKCDS).IXRRAT, replaces link to RLNODE named NODOLD by one to RLNODE named NODNEW. A delay with parameters RLDPAR (DLYNEW) is assigned the new link.

Assembler Inputs

Arguments.

NODOLD - terminator of return flow link to be replaced

NODNEW - terminator of new return flow link

DLYNEW - delay assigned to new link

Parameter Slots. None.

Examples

CHRLK can be used to change the return flow path to a node.

NODEA.

```
ACSUP (2 = STGBR (1 = RLRLK (P = * NODEA, 1.), + R1 FROM SUPTA
                      RLRLK (P = * SUPTA, 3.), + R1 FROM CONUS
                      RLDLK (P = * CONUS, 0.)$
                  3 = CHRLK (P = * SUPTA, * SUPTB, 5.))) + R1 FROM
                      CONUS
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

IXRRAT - index of element in LKDST (LKCDS) corresponding to return flow path of current interest

RRATE - return flow rate associated with return link of current interest

Verb Inputs

From Calling Program. Arguments - see above

From RLLNK. None.

Verb Outputs

To RLLNK (1st call)

Arguments:

NODOLD - terminator of return flow link to be replaced

IXRRAT - index of element in dataset corresponding to return flow path of current interest

-3 - substitute new link for an existing one

0 - indicate there is to be no delay

To RLLNK (2nd call)

Arguments:

NODNEW - terminator of new return flow link

IXRRAT - index of elemnt in dataset corresponding to return flow path of current interest

RRATE - return flow rate associated with return link of interest

DLYNEW - number of rate level delay type associated with the link

Programs Called

Verbs. None.

Other. RLLNK.

Input/Ouput Files Used

NPRNT - Print File

(CHRLK-2)

DEDLK (NODOLD)

General Description

This verb is used to delete a demand link to NODOLD.

Deletes link to RLNODE named NODOLD from demand path pointed to from LKDST (LKCDS).IXDRAT.

Assembler Inputs

Arguments.

NODOLD - terminator of demand link to be deleted

Parameter Slots. None.

Examples

A demand link to SUPTA is to be deleted in the 3rd stage period at NODEA. NODEA.

```
ACSUP (2 = STGBR (1 = RLRLK (P = * NODEA, 3.), + R1 FROM SUPTA
                      RLDLK (P = * SUPTA, 1.)$ + D1 TO SUPTA
                  3 = DEDLK (P = * SUPTA)))    + DELETE D1 LINK
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

IXDRAT - index of elements in LKDST (LKCDS) corresponding to demand flow paths of current interest

Verb Inputs

From Calling Program. Argument - see above

From RLLNK. None.

Verb Outputs

To RLLNK (1st call)

NODOLD - terminator of demand link to be deleted

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

0 - zero flow rate assigned to the link

0 - indicates there is no delay

(DEDLK-1)

To RLLNK (2nd call)

NODOLD - terminator of demand link to be deleted

IXDRAT - index of element in dataset corresponding to demand flow path of current interest

-1 - delete link

0 - indicates there is no delay

Programs Called

Verbs. None.

Other. RLLNK.

Input/Output Files Used

NPRNT - Print Files

(DEDLK-2)

DERLK (NODO)

General Description

This verb deletes a return link to a specified node during a model run.

Deletes link to RLNODE named NODO from return flow path pointed to by LKDST(LKCDS).IXRRAT.

Assembler Inputs

Arguments.

NODO - terminator of link to be deleted from return flow path.

Parameter Slots. None.

Examples

A return path to NODEA is to be deleted in stage 5 following the deletion of the demand path in stage 3.

```
NODEA.
     ACSUP (2 = STGBR (P = 7 $
                       1 = RLRLK (P = *NODEA, 3.),
                           RLDLK (P = *SUPTA, 2.) $
                       3 = DEDLK (P = *SUPTA) $
                       5 = DERLK (P = *NODEA )))
```

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/

IXRRAT - index of element in LKDST (LKCDS) corresponding to return flow path of current interest

RLNNAM - name of system dynamics node

In /LC/

RLC - unique set of PDS coordinates for datasets that represent or own system dynamics flow paths

In /SUPC/

IXRLC - index to the coordinates of the LKDST (LKCDS) dataset in RLC

Verb Inputs

From Calling Program

Argument - see above.

From RLLNK. None.

Verb Outputs

To RLLNK (1st call):

Arguments -

NODOLD - terminator of return flow link to be deleted

IXRRAT - index of element in dataset corresponding to return flow path of current interest

0 - zero flow rate assigned to link

0 - indicates there is no delay.

To RLLNK (2nd call):

Arguments -

NODOLD - terminator of return flow link to be deleted

IXRRAT - index of element in dataset corresponding to return flow path of current interest

-1 - delete link

0 - indicates there is no delay.

Programs Called

Verbs. None.

Other. RLLNK.

Input/Output Files Used

NPRNT - Print file.

(DERLK-2)

DLEVS

General Description

Verb to print current levels in rate/level delays. An example of the report printed is shown in the accompanying figure.

Assembler Inputs

Arguments.

None.

Parameter Slots.

None.

Examples

The verb DLEVS is generally included in the node ZRPRT after the verb ORLD. This combination provides a listing of the rate/level delay parameters and the values in each rate/level delay.

Statistics Collected

None.

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RLDST(12) - PDS dataset type name RLD

RL(1) - IXDPAR, 1st element of system dynamics delay of current interest

RLDPAR(2,J) - parameters for J-th system dynamics delay, mean delay time for 1st and 3rd order delay; points to DTABLE for boxcar delays

RLDTYP - array of RL delay type names

In/DPTRS/:

DPTRS - pointers into DTABLE

Verb Inputs

From Calling Program.

None.

From IDSLON.

Arguments -

First - index in DSPOOL of first element of RLD dataset

Third - number of elements in the RLD dataset

Verb Outputs

To File NPRNT:

Current levels in RL delays report

To IDSLON.

Arguments -

Second - name of PDS dataset to loop on

Programs Called

Verbs. None

Other. DLYLEV, IDSLON, LIN

Input/Output Files Used

NPRNT - Print File

** MAWLOGS MODEL DSS4 **

ANALYST., BRM
DATE. 10/ 18/ 1975
RUN NAME.FEAR SLICE
TNOW 159.980

SUMMARY OUTPUT

CONTENT OF RATE/LEVEL DELAYS

FLD COORD	TYPE	PARAMETER	CURRENT LEVEL	DELAY NUMBER
1	RLMD1	7.000	6.91866E+05	36
2	RLMD3	44.300	5.59172E+06	4
3	RLMD3	19.000	5.60160E+02	18
4	RLMD1	5.300	8.06202E+02	35
5	RLMD3	15.700	2.24889E+03	3
6	RLMD1	6.000	1.21321E+00	14
7	RLMD3	9.000	0.	17
8	RLMD1	7.000	8.25343E+05	36
9	RLMD3	44.300	1.50397E+07	4
10	RLMD3	19.000	8.02874E+03	18
11	RLMD1	5.300	2.01060E+04	35
12	RLMD3	15.700	3.35801E+05	3
13	RLMD1	6.000	5.49507E+01	14
14	RLMD3	9.000	8.20560E+02	17
15	RLMD1	7.000	8.63185E+05	36
16	RLMD3	44.300	8.65122E+06	4
17	RLMD3	19.000	9.83650E+02	18
18	RLMD1	5.300	5.64049E+04	35
19	RLMD3	15.700	1.45969E+05	3
20	RLMD1	6.000	3.23886E+01	14
21	RLMD3	9.000	0.	17
52	RLMD1	7.000	8.09268E+05	36
53	RLMD3	44.300	6.70768E+06	4
54	RLMD3	19.000	6.38873E+02	18
55	RLMD1	5.300	5.13517E+04	35
56	RLMD3	15.700	1.35701E+05	3
57	RLMD1	6.000	7.88506E+01	14
58	RLMD3	9.000	3.05236E+01	17
59	RLMD1	7.000	4.65610E-02	36
60	RLMD3	44.300	1.73123E+07	4
61	RLMD3	19.000	8.02874E+03	18
62	RLMD1	5.300	3.80279E+04	35
63	RLMD3	15.700	1.12043E+05	3
64	RLMD1	6.000	1.56992E+01	14
65	RLMD3	9.000	5.77621E-01	17
66	RLMD1	7.000	6.39261E+05	36
67	RLMD3	44.300	1.32275E+07	4
68	RLMD3	19.000	2.03845E+03	18
69	RLMD1	5.300	7.67101E+03	35
70	RLMD3	15.700	2.36220E+04	3
71	RLMD1	6.000	4.09689E+00	14
72	RLMD3	9.000	0.	17
73	RLMD1	7.000	1.05227E+06	36
74	RLMD3	44.300	6.87347E+06	4
75	RLMD3	19.000	5.60183E+02	18
76	RLMD1	5.300	1.39157E+03	35
77	RLMD3	15.700	3.89847E+03	3
78	RLMD1	6.000	2.00606E+00	14
79	RLMD3	9.000	2.55009E-04	17
80	RLMD1	7.000	2.31815E-03	36
81	RLMD3	44.300	1.48903E+07	4

FIGURE - EXAMPLE OF REPORT FROM DLEVS

THIS PAGE LEFT INTENTIONALLY BLANK

DMPLC

General Description

Dumps contents of Rate/Level COMMON /LC/ for debugging purposes.

Assembler Inputs

Arguments. None.

Parameter Slots. None.

Examples

DMPLC can be placed in a separate subnode of ZINIT so that it can be executed through a GASP exogenous event of type -8.

```
ZINIT.
      /7/  DMPLC     + DUMP CONTENTS OF /LC/
```

GASP Files Used

None.

Permanent Attributes Accessed

In /DSTOR/:

DSPOOL - storage array for permanent datasets

In /RLSYS/:

RLDST(6) - LDSC - 6th PDS dataset type name

In /LC/:

RLC(*,I) - Ith unique set of PDS coordinates for datasets that represent or own system dynamics flow paths

LCIX(*) - array of pointers into RLC

NRLDIM - maximum number of coordinates permitted in an entry in RLC (1st dimension of RLC)

NRLC - maximum number of entries permitted in RLC (2nd dimension of RLC)

NLCIX - dimension of LCIX

NXTRLC - index of next available position in RLC

NXTLCX - index of next available position in LCIX

In /TBLCOM/:

DTABLE - array for storing pairs of values, in this case names of rate dataset types and pointers to LCIX

Verb Inputs

From Calling Program. None.

From IDSLON.

Arguments -

I - index of first element of LDSC dataset in DSPOOL

NEL - number of elements

Verb Outputs

To IDSLON.

Argument -

LDSC - name of dataset to loop on

To Calling Program. None.

Programs Called

Verbs. None.

Other. GETSET, IDSLON, IUNPKZ, PERMDS.

Input/Output Files Used

NPRNT - Print File.

IBXST

General Description

Initializes statistics collection of boxcar trains.

Initializes OBOX statistics datasets. Assumes input results in content.

1 NMAX - Number of cells for data collection

3 XW Width of cell or zero

IBXST initializes OBOX to

1 Packs IOLD=1 with NMAX

2 Sets INEW=NCUR=1, Packed

3 If XW.LE.0, Sets XW=1

4 Sets TOLD=TWARM

5 Sets FOLD = -1

6 - (NMAX+5) Clears to zero

Checks NMAX against NEL and sets NMAX = NEL - 5

Assembler Inputs

Arguments. None

Parameter Slots. None

Examples

IBXST should be included in node ZINIT to be executed at beginning of a model run.

ZINIT. ..., IBXST, ...

GASP Files Used

None.

Permanent Attributes Accessed

OBOX - All order ship time measuring boxcar datasets

(IBXST-1)

Verb Inputs

From Calling Program.

In/RLSYS/:

TSTEP - size of time step

In/GSPCOM/:

TBEG - beginning time of model run

In OBOX Dataset:

1 - number of cells for data collection

3 - width of cells or zero

From IDSLON.

Arguments -

First - index is DSPOOL of first element in OBOX dataset

Third - number of elements in dataset

Verb Outputs

To IDSLON.

Arguments -

SECOND - name of dataset to loop on

To Calling Program.

In OBOX Dataset:

1 Packs IOLD=1 with NMAX

2 Sets INEW=NCUR=1, Packed

3 If XW.LE.0, Sets XW=1

4 Sets TOLD=TWARM

5 Sets FOLD = -1

6 - (NMAX + 5) Clears to zero

Checks NMAX against NEL and sets NMAX = NEL - 5

Programs Called

Verbs. None

Other. CLEAR, IDSLON, IPAKZ

Input/Output Files Used

NPRNT - FORTRAN logical file name for printed reports

IRLD

General Description

Standard input verb to initialize rate/level delays. Both the type of delay and the parameters for the delay must be specified in the input deck. Card formats are shown in the attached table. Two card types are used. Type IRLD1 is used, one per delay, to identify the delay form and associate with it a unique index which can be referenced from the model description or other datasets. The second card type, IRLD2, is used to describe an arbitrary delay time distribution when the distribution is a "boxcar" as defined by Forrester. For a boxcar delay, the IRLD2 cards must immediately follow the IRLD1 card. The order in which delays are defined within the deck is immaterial. A card containing ENDRLD in card columns 1-6 signals the end of the IRLD deck.

The PDS dataset of type RLD is defined in this module. RLD datasets are then created as needed during a model run.

Assembler Inputs

Arguments.

None.

Parameter Slots.

None.

Examples

The verb IRLD should be included in the initialization node of a model after the PDS input verbs such as IPDS and RLPDS since IRLD calls MAKPTR.

ZINIT. ISTAT, RLPDS, IPDS, IRLD, ...

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RLDNAM - array of rate/level delay names

RLDPAR - array of rate level delay parameters

(1,•) - delay type

(2,•) - delay parameter

Verb Inputs

From Calling Program.

None.

From File NCRDR.

Delay types and parameters

From INDTB.

Argument -

IXDTI - index to first entry in new table in DTABLE.

Verb Outputs

To RLDRPT.

In/RLSYS/:

RLDPAR - parameters of rate/level delays

RLDNAM - names of rate/level delays

To DEFDST.

In/BUF/:

BFF - characteristics to define RLD dataset type

To INDTB.

Arguments -

DT - array of table entries to be placed in DTABLE

IT7 - number of entries in new table

To Calling Program.

None.

Programs Called

Verbs. None

Other. DEFDST, INDTB, INEROR, MAKPTR, RLDRPT

Input/Output Files Used

NCRDR - file to read card input

NPRNT - file for printed output

TABLE IRLD DATA CARD FORMATS

FORMAT FOR CARD TYPE IRLD1

COLUMNS	FORTRAN FORMAT	DESCRIPTION
1-5	A5	"IRLD1", CARD TYPE IDENTIFIER
6	1X	IGNORED
7-8	I2	DELAY INDEX (INDEX BY WHICH REFERRED TO IN MODEL DESCRIPTION)
9	1X	IGNORED
10-14	A5	DELAY TYPE CODE (SEE NOTE 1)
15	1X	IGNORED
16-25	F10.3	IF NOT BOXCAR, MEAN DELAY. IF BOXCAR, IGNORED.
26-79	9A6	DELAY DESCRIPTION
80	1X	IGNORED

FORMAT FOR CARD TYPE IRLD2

COLUMNS	FORTRAN FORMAT	DESCRIPTION
1-5	A5	"IRLD2", CARD TYPE IDENTIFIER
6	1X	IGNORED
7-8	I2	DELAY INDEX (MUST MATCH THAT IN PRECEDING IRLD1 CARD)
9	1X	IGNORED
10-15	F6.0	DELAY TIME ASSOCIATED WITH 1ST PROB
16-19	F4.4	PROB OF 1ST DELAY TIME INCREMENT (OR 8TH, 15TH,...)
20-25	F6.0	2ND DELAY TIME INCREMENT
26-29	F4.4	PROB OF 2ND DELAY TIME INCREMENT (OR 9TH, 16TH,...)
.		
.		
.		
70-75	F6.0	7TH DELAY TIME INCREMENT
76-79	F4.4	PROB OF 7TH DELAY TIME INCREMENT (OR 14TH, 21ST)

AS MANY POINTS AS DESIRED MAY BE SPECIFIED. A CARD CONTAINING "ENDDT" IN COLUMNS 1-5 SIGNALS THE END OF A SET OF IRLD2 CARDS. AN ENTRY WITH PROB. LE.O. SIGNALS THE END OF A TABLE WITHIN A CARD. DELAY TIME INCREMENTS ARE ASSUMED INPUT IN ORDER OF INCREASING AGE WITHIN THE DELAY.

NOTE 1 - RATE/LEVEL DELAY FORMS

CODE NAME	DESCRIPTION
RLID1	1ST ORDER EXPONENTIAL, INFORMATION
RLID3	3RD ORDER EXPONENTIAL, INFORMATION
RLIDB	ARBITRARY (FORRESTER'S BOXCAR), INFORMATION
RLMD1	1ST ORDER EXPONENTIAL, MATERIAL
RLMD3	3RD ORDER EXPONENTIAL, MATERIAL
RLMDB	ARBITRARY (FORRESTER'S BOXCAR), MATERIAL
ENDDT	END OF SUBDECK OF IRLD2 CARDS FOR A BOXCAR DELAY

(IRLD-3)

IZRL

General Description

IZRL is the restart input module for system dynamics data structures. It reads the file created by SVRL. If the data encountered on the restart file are not introduced by the label "SVRL", the run is stopped.

The data are read as seven logical records in binary mode. These records are described in the accompanying tabulation.

Record Number	Description
1	"SVRL", the data identifier
2	TSEP, RLDST, TNXT, RLNNAM, NRLN, NRLNMX, RL, RLDPAR, NRLDPR, RLDNAM, DLRIXK, RLDTYP, RATDST, NRATDT, i.e., variables in /RLSYS/
3	KRLST, KRLST1, NRLST, NXRLST, RLSTIU, RLSTIS, i.e., variables in /RLSYS/
4	NRLDIM, NRLCSV, NLCSV, NXTRLC, NXTLCX, NFRELC, i.e., variables in /LC/
5	RLC - set of PDS coordinates for datasets that represent system dynamics flow paths
6	LCIX - array of pointers into RLC
7	RLSNAM in COMMON/STNAM/

Assembler Inputs

Arguments. None.

Parameter Slots. None.

Examples

IZRL should be included in subnode 2 of the node ZREST.

```
ZREST.  SVRUN (1 = SVPDS, SVRL, SVSTA)
  /2/   IZRUN (1 = IZPDS, IZRL, IZSTA)
```

(IZRL-1)

<u>GASP Files Used</u>

None.

<u>Permanent Attributes Accessed</u>

In/RLSYS/: see record numbers 2 and 3 in table above.

In/LC/: all variable except NRLC, NLCIX, LRLCHG, and LRLPTR

In/STNAM/: RLSNAM

<u>Verb Inputs</u>

<u>From Calling Program</u>.

None.

<u>Verb Outputs</u>

<u>To Calling Program</u>

None.

<u>To Permanent Attributes</u>

Initial values read from a Restart File for all permanent attributes accessed.

<u>Programs Called</u>

<u>Verbs</u>. None

<u>Other</u>. RLSRPT

<u>Input/Output Files Used</u>

NFREST - restart input file

NPRNT - print file

(IZRL-2)

MAPLK

General Description

Maps flow rates and in-transit resources from one flow network to a revised form of it. Two kinds of mappings are performed. One kind concerns the disposition of flows and resources in links that have been deleted. The other concerns a redistribution of flows and resources specified by datasets of type NODMAP. A particular execution of MAPLK may involve only one type of mapping, or it may involve combinations of the two types.

When the first kind of mapping is performed, i.e., the kind necessary to adjust for revisions to the network structure, the possibilities are limited to combining resources in transit in a deleted link with resources in transit to its successor in the same flow path. If a successor does not exist, no transfer is made. Whether any transfer is necessary or possible in a flow path or not, MAPLK carries out the final step of converting the revised flow path to reference form. That consists of deleting the old path from IZHOLD, deleting the pointer entry in IZHOLD, and setting the PDS path-owner element to point to the new path representation in DTABLE.

The second kind of mapping concerns the transfer of flow rates and resources from one node to one or more others. A NODMAP dataset specifies the fraction to be transferred from its owner node (its coordinate) to one or more nodes (one of which may be itself).

Assume that it is desired to assign half the network flows currently associated with node A to node AA, leaving the remainder with node A. Then a NODMAP (A) dataset specifying fraction .5 for AA and .5 for A, in that order, is required. Mappings are defined only between corresponding paths and nodes. That is, flow in the kth demand path of dataset type TYP of node A may only be mapped to other kth demand paths owned by TYP datasets. Further, within corresponding paths the mapping is restricted to corresponding links or, in the absence of direct correspondents, correspondents

of the successor of the original link in the path of origin. (The latter rule is the same as that described above for mapping to reflect structural changes only.)

No rates or resources are transferred if corresponding paths and links or successor links cannot be found. As currently written, only path-owner dataset types EXRAT and FCRAT are accommodated.

An outline of the operation of MAPLK is shown in the accompanying figure. Details of the data structure on which it operates are given in Chapter III.

Assembler Inputs

Arguments. None.

Parameter Slots. None.

Statistics Collected

None.

Examples

MAPLK was designed for use in PS1 of vRLFLK. Its use is mandatory if there are changes to the flow network or if there are NODMAP datasets. Its place in the system dynamics control node is shown below.

```
SDYNC.RLNOD (P = *SDYNC, n),
      RLCTR (1 = ... $
             2 = RLSU (1 = RLFLK (1 = MAPLK )), FADE ),
      *(next node in system dynamics node cycle)
```

GASP Files Used

None.

Permanent Attributes Accessed

PDS Elements.

(RATDST(IRATDT)) (coords).iel - all flow-path-owning pointers in model

NODMAP (node) - all NODMAP datasets

RLD (dlycoord) - RLD datasets whose contents are to be mapped from one link to another

(MAPLK-2)

In /TBLCOM/:

DTABLE - flow paths affected by mappings.

Verb Inputs

From Calling Program. Flow network in reference form except for flow paths that were changed. These are to be in DTABLE, and all LDSC-based membership data are to be in reference form, but the pointer to the DTABLE-resident revised path is to be in IZK of a pointer entry in IZHOLD, while the IZKC element of the pointer entry points (neg) to the pushdown stack containing the original flow path. The path-owner PDS element is a negative index of the pointer entry in IZHOLD.

From IDSLON (I, RATDST (IRATDT), NEL). Arguments. I - if >0, index in DSPOOL of 1st element of current RATDST(IRATDT) dataset; if ≤0, signal that all RATDST(IRATDT) datasets have been accessed; NEL - number of elements in current dataset. Also COORD in /DPTRS2/, coordinates of current dataset.

From PERMDS (GETSET, NODMAP, COORD(1), 0, NM). Argument NM, elements of NODMAP (COORD(1)) if it exists.

In /DSNOW/ -

IXDST - >0 if NODMAP dataset type defined, ≤0 if not

IXELI - >0 if NODMAP (COORD(1)) exists, ≤0 if not

From PERMDS (VALEL, RATDST (IRATDT), RATCD, K, ZHPN).

Argument ZHPN, pointer to revised flow path.

In /DSNOW/ - IXELI, <0 if element K of current dataset does not exist

From 1st ZREMEN. In /ZSTACK/ - entry removed from pushdown stack of old flow path

IZREF - >0, index in IZHOLD of next entry in path; ≤0, signals last entry

IZKC - node and RLD coordinate

IZK - flow rate

(MAPLK-3)

From PERMDS (RELSET, RLD, FLOAT (IDLYO)). None.

From 2nd ZREMEN. None. (Releases pointer entry in IZHOLD.)

From IRSTK (INOD, IPTRWN). Function value, INEW, location in "new" path of node corresponding to current node in "old" path, or ≤0 if there is no correspondent in new

From IRSTK (MAPTO, IPTRWN). Function value, INEW, location in "new" path of node corresponding to MAPTO node, or ≤0 if node cannot be found

From IRSTK (INODS, IPTRWN). Function value, INEW, location in "new" path of node corresponding to successor of current node in "old" path, or ≤0 if INODS cannot be found

From DMAP (FLOAT (IDLYO), FLOAT (IDLYN), AD). None.

Verb Outputs

To Permanent Attributes.

(RATDST(IRATDT)) (coords).iel - all flow-path-owning pointers to flow paths that were revised have been converted to reference form

RLD (dlycoord) - RLD datasets subject to mapping will have altered content; some may have been deleted if they belonged to an altered path

To Calling Program. Flow network fully converted to reference form, with flow rates and in-transit resources redistributed as specified by any NODMAP datasets

To IDSLON (I, RATDST (IRATDT), NEL). Argument RATDST (IRATDT), dataset type to be accessed. Also IPDRET in /PDSRET/, return address in MAPLK.

To PERMDS (GETSET, NODMAP, COORD(1), 0, NM). Arguments specifying dataset NODMAP owned by node that owns current RATDST (IRATDT) flow-path-owning dataset

To PERMDS (VALEL, RATDST (IRATDT), RATCD, K, ZHPN). Arguments specifying RATDST (IRATDT) dataset for current destination node of current NODMAP, and K, the pointer element for the current flow path

(MAPLK-4)

To 1st ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of entry to be removed from old flow path stack

To PERMDS (RELSET, RLD, FLOAT (IDLYO)). Arguments to cause release of RLD dataset associated with link in old flow path that has been deleted by preceding ZREMEN

To 2nd ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of pointer entry for old and new paths, which can now be deleted because old path has been released from IZHOLD and new path has been in DTABLE since vRLFLK

To IRSTK (INOD, IPTRWN). Arguments:

- INOD - node corresponding to current node map source path to be sought in current map destination path
- IPTRWN - pointer to current map destination path in DTABLE

To IRSTK (MAPTO, IPTRWN). Arguments:

- MAPTO - current destination node in current NODMAP dataset, to be sought in current map destination path
- IPTRWN - pointer to current map destination path in DTABLE

To IRSTK (INODS, IPTRWN). Arguments:

- INODS - successor node to current node in current map source flow paths to be sought in current map destination flow path
- IPTRWN - pointer to current map destination path in DTABLE

To DMAP (FLOAT (IDLYO), FLOAT (IDLYN), AD). Arguments:

- FLOAT (IDLYO) - coordinate of RLD dataset to serve as source of current delay mapping
- FLOAT (IDLYN) - coordinate of RLD dataset to serve as destination of current delay mapping
- AD - fraction of source delay to be mapped to destination delay.

Programs Called

Verbs. None.

(MAPLK-5)

Other. DMAP, GETSET, IDSLON, IRSTK, IUNPKZ, PERMDS, RELSET, VALEL, ZSTAK.

Input/Output Files Used

NPRNT - print file, for KTRACE and error message.

(MAPLK-6)

Loop through rate (i.e., flow-path-owning) dataset types in RATDST

- For each type loop (IDSLON) through each dataset
  - For each dataset
    1. Obtain NODMAP, if any, of owner node and set NMAP = .TRUE. if no NODMAP for owner, leave NMAP = .FALSE.
    2. Loop through path owner elements of dataset
        1. if the element is zero, skip to next
        2. if the path is in reference form and there is no NODMAP mapping, skip to next
        3. if the path is not in reference form, set OLDSTK to 1 if old path is in IZHOLD, set OLDSTK to 2 if path is in DTABLE
        4. if there is a NODMAP mapping, set up loop through nodes in NODMAP dataset; if no NODMAP mapping, set up single pass loop to map all to itself
        5. Loop through each NODMAP node or make single pass; for each pass
            - Loop through each element of old path, transferring specified fraction of rate and delay content to corresponding entry in corresponding path of current NODMAP node; for single pass (no NODMAP), transfer to revised version of current old path
        6. if old path was in IZHOLD, release it and associated RLD datasets, and release intermediate pointer entry after setting DTABLE pointer into current path owner element

Retlog

Outline of MAPLK Logic

(MAPLK-7)

THIS PAGE LEFT INTENTIONALLY BLANK

MLOOP

General Description

Loop on list of management classes in MGTCL (RLNOD), executing PSI for each. Current management class is set into CLASM in /SUPC/.

Assembler Inputs

Arguments. None.

Parameter Slots.

1 - verbs to be executed for each management class

Examples

MLOOP can be used at a node that handles more than one management class, to process all the classes -

```
MLOOP (1 = SETPR (P = 2.),
           CSUP (P = 0  $                 + SUPPLY PROCESSING
                 1 = LARPL (P = 0, 1), + REPLENISHMENT
                     PLOOP (1 = CRCPT (P = 0), + DIVERSION
                             UDRCT (P = -2,0))))
```

GASP Files Used. None.

Permanent Attributes Accessed

PDS Dataset MGTCL (node) - management class codes for this node

In /RLSYS/:

RLNOD - current RL node number

RLDST(7) - MGTCL, dataset type name

In /DSNOW/:

NE - number of elements in dataset of current interest

IXEL1 - index in DSPOOL of first element in dataset of current interest

Verb Inputs

From Calling Program. None.

From PSI. None.

Verb Outputs

To Permanent Attributes

In /SUPC/:

NM - number of management classes

ZMCL - dataset of management classes

To PSI.

In /SUPC/:

CLASM - current management class

Programs Called

Verbs. None.

Other. GETSET, PERMDS.

Input/Output Files Used

NPRNT - Print File

NODMP (MFILE)

General Description

This verb merges or maps the rate flows and levels of materiel from one node to one or more other nodes in a model. This may be done when a node is deactivated or aggregated with another node. This module reads node mapping data and stores it in datasets of type NODMAP for use by model change logic. The coordinate of a NODMAP dataset is the RL node number of an old node, i.e., one previously or currently active. The content of a dataset is a list of pairs of the form (NEWNODNUM,F), where F is the fraction of a rate or level belonging to the old node that is to be transferred to the node whose rate/level number is NEWNODNUM. The new node may be one currently active or one about to be activated. The terms old and new refer primarily to ownership of the rates and levels, and not to when the node has been or will be active. Thus, a NODMAP dataset maps the content of one node into one or more other nodes. Data are read from file MFILE. Nodes are identified by name in the input cards. The card format is given in the following table. Verb RELNM may be used to release the NODMAP datasets at the conclusion of the changes to be implemented at the present stage.

Assembler Inputs

Arguments.

MFILE - file number from which NODMP data cards are read.

Parameter Slots. None.

Examples

This verb is generally placed in a subnode of ZINIT so it may be scheduled by an exogenous event. A RELNM verb should also be scheduled to release the NODMAP datasets once the mapping is created.

```
ZINIT.
    /3/ NODMP (P = 5), CHPDS (P = 5), CHRLD (P = 5), RLSMU,
        *RLCTR
```

(NODMP-1)

```
RLCTR.RLCTR (1 = TSTEP (1 = DELAY (P = 0), *RLCTR) $
                          2 = RLSU (1 = RLFLK (1 = MAPLK), + MAP LINK CONTENT
                                              RELNM),    + RELEASE NODE MAP
                                 FADE )
```

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/:

RLNNAM - rate/level node names array

Verb Inputs

From Calling Program. None.

From File MFILE.

See accompanying table.

Verb Outputs

To Permanent Attributes.

PDS Datasets:

NODMAP (node) - new node names and fractions to determine mapping

To RLNIX.

Argument - rate/level node name for translation to rate/level node number

To IXZNOD.

Argument - rate/level node name for translation to IZNODE number.

To Calling Program. None.

Programs Called

Verbs. None.

Other. IDSLON, INEROR, IXZNOD, LIN, MSHIFT, PERMDS, RLNIX.

Input/Output Files Used

None.

(NODMP-2)

TABLE - FORMAT FOR CARDS READ FROM MFILE BY VERB NODMP

CARD COLS	FORTRAN FORMAT	CONTENT
1 - 5	A5	NODMP - CARD NAME
6 -10	5X	IGNORED
11-16	A6	RL NODE NAME OF OLD NODE, OR ?***END? IF END OF DECK
17-20	4X	IGNORED
21-26	A6	RL NODE NAME OF 1ST NEW NODE
27-30	F4.4	PROPORTION OF OLD TO BE ASSIGNED TO 1ST NEW
31-36	A6	RL NODE NAME OF 2ND NEW NODE
37-40	F4.4	PROPORTION OF OLD TO BE ASSIGNED TO 2ND NEW
.		
.		
.		
71-76	A6	RL NODE NAME OF 6TH NEW NODE
77-80	F4.4	PROPORTION OF OLD TO BE ASSIGNED TO 6TH NEW

MORE CARDS MAY BE USED IF MORE THAN SIX NEW NODES ARE TO BE ACCOMMODATED. PDS PARAMETER MAXELS IN /PDSSYS/ IS THE ONLY LIMITING FACTOR ON THE NUMBER OF NEW NODES FOR ONE OLD. IN PARTICULAR, (MAXELS-1)/2 IS THE LIMIT.

(NODMP-3)

THIS PAGE LEFT INTENTIONALLY BLANK

ORLD

General Description

Prints Rate/Level Delays. This verb is designed so that the routine RLDRPT can be referenced in a model description.

Assembler Inputs

Arguments. None.

Parameter Slots. None.

Examples

Verb ORLD can be included in the first subnode of node ZRPRT to provide a listing of rate/level delays and values.

ZRPRT. . . ., ORLD, . . .

GASP Files Used. None.

Permanent Attributes Accessed. None.

Verb Inputs

From Calling Program. None.

Verb Outputs. None.

Programs Called

Verbs. None.

Other. RLDRPT.

Input/Output Files Used

NPRNT - Print File

THIS PAGE LEFT INTENTIONALLY BLANK

PLOOP

General Description

Loop on priorities listed in PRIORY (RLNOD), execute PS1 for each. Set priority value into PRI in /SUPC/. Also sets index to priority in IPRI and CLAS * 100 + PRI in CLASPR, both in /SUPC/.

Assembler Inputs

Arguments. None.

Parameter Slots.

1 - verbs to be executed for each priority listed in PRIORY dataset at the node.

Examples

PLOOP is used to set up a loop on all priorities of materiel at a node.

```
CSUP (P = 1  $
        1 = PLOOP (1 = PRSPL,
                        CDMDS . . . ,    + PROCESS DEMANDS
                        CRCPT . . . ,    + PROCESS RECEIPTS
                        UDRAT . . . ))   + UPDATE RATES
```

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

RLNOD - current RL node number

RLDST - array of dataset type names

LKCDS(2) - 2nd coordinate of PDS dataset of type LKDST whose flow paths are of current interest

IXRRAT - index of element in LKCDS corresponding to return flow path of current interest

In /DSNOW/:

NE - number of elements in dataset of current interest

IXEL1 - index in DSPOOL of first element in dataset of current interest

In /SUPC/:

RATPS(1) - value of the first element of dataset type FCRAT and coordinate CLASPR

C2 - coordinate array (node, claspr)

Verb Inputs

From Calling Program. None

From PS1. None.

Verb Outputs

To Permanent Attributes. None.

To PS1.

In /SUPC/:

PRI - priority code of materiel flow of current interest

CLASPR - CLAS * 100 + PRI

C2(2) - CLASPR

In /RLSYS/:

LKCDS(2) - 2nd coordinate of PDS dataset of current interest

IXRRAT - index of element in LKCDS corresponding to return flow path of current interest

Programs Called

Verbs. None.

Other. CLASS, GETSET, PERMDS, VALEL.

Input/Output Files Used

NPRNT - Print File

PRYBR

General Description

Branches on priority. The PS implemented is IPRI, the priority index in /SUPC/.

Assembler Inputs

Arguments. None.

Parameter Slots.

IPRI - up to 20 parameter slots can be called, one for each priority index in the model

Examples

PRYBR can be used to set up different logical paths for different priorities:

```
CDMDS (P = 1 $              + PROCESS DEMANDS
       1 = WBFIL $          + DEMAND SATISFACTION
       2 = PRYBR (1 = CBOP $  + BO/PASS HIPRI NOFILL
                  2 = CBO ))  + BO LOPRI NOFILL
```

GASP Files Used

None.

Permanent Attributes Accessed

None.

Verb Inputs

From Calling Program.

In /SUPC/:

IPRI - index of priority under consideration

Verb Outputs

None.

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

NPRNT - Print File

THIS PAGE LEFT INTENTIONALLY BLANK

RELNM

General Description

Releases all node map datasets. IDSLON is used to loop on NODMAP datasets, placing their coordinates in /DPTRS/. A RELSET operation is then used to release each NODMAP dataset.

Assembler Inputs

Arguments.

None.

Parameter Slots.

None.

Examples

RELNM should be executed after a MAPLK so that NODMAP datasets are released after the mapping is accomplished.

```
RLCTR (1 = TSTEP (1 = DELAY (P = 0), *RLCTR)$
          2 = RLSU (1 = RLFLK (1 = MAPLK),
                        RELNM),
              FADE)
```

GASP Files Used

None.

Permanent Attributes Accessed

In /DPTRS/:

DPTRS - COORD, coordinate of dataset whose dataset type name is NODMAP

Verb Inputs

From Calling Program.

None.

From IDSLON.

In/DPTRS/:

DPTRS - COORD - coordinates of dataset whose dataset type name is NODMAP

Verb Outputs

To IDSLON.

Arguments -

Second - name of NODMAP dataset to loop on.

Programs Called

Verbs. None

Other. IDSLON, PERMDS, RELSET

Input/Output Files Used

NPRNT - Print File

RLCTR

General Description

Principal control module for system dynamics simulation in MAWLOGS. Initiates the node cycles necessary to update levels and rates, and the setup mode cycles used to define and modify the flow path network. The first execution in a model run started from scratch is automatically in setup mode. A flowchart of the control logic is shown in the accompanying figure. On initial entry, RLSW is set to LEVELS and a cycle through the system dynamics nodes is executed. Since the control node, of which RLCTR is the principal member, is in the cycle, this leads to execution of a second cycle with RLSW = RATES. When RLCTR is entered at the end of the RATES cycle, RLSW is set to NEXT and a branch is made to PS1 or PS2 depending on the value of RLMODE. PS2 is used to conclude a setup pass with such actions as converting flow path representations to reference for (vRLFLK) and releasing NODMP datasets (vRELNM) if any were used. PS1 is used to schedule the next levels and rates update at the end of the next time step (vTSTEP). In either case control is relinquished to MAWGSP.

Assembler Inputs

Arguments. None.

Parameter Slots.

1 - Schedule next time step (vTSTEP)

2 - Concluding actions of setup pass (vRLFLK)

Examples

If SDCTR is the system dynamics control node, a typical setup for RLCTR is

```
SDCTR.RLNOD (*SDCTR, 4.),
      RLCTR (1 = TSTEP (1 = DELAY (P = 0), * SDCTR) $
             2 = RLSU (1 = RLFLK (1 = MAPLK)),
                 FADE ),
     *EXOGD  $
```

(RLCTR-1)

Note that FADE is used to conclude PS2. EXOGD is the first node in the system dynamics node cycle - often it is one that generates exogenous demand on the system. The first execution of SDCTR is typically scheduled via vFILE in nZINIT.

GASP Files Used

The time file is used to schedule the next levels and rates updates.

Permanent Attributes Accessed

In /RLSYS/ -

RLSW - rates/levels indicator

RLMODE - simulation of setup indicator

RATES

LEVELS - RLSW values

NEXT

SIM

\- RLMODE values

SETUPM

Verb Inputs

From Calling Program.

In /RLSYS/ -

RLMODE - SETUPM if setup pass is desired (use vRLMSU)

Verb Outputs

In /RLSYS/ -

RLSW - value for current cycle

RLMODE - SIM if it was SETUPM at start of current execution

Programs Called

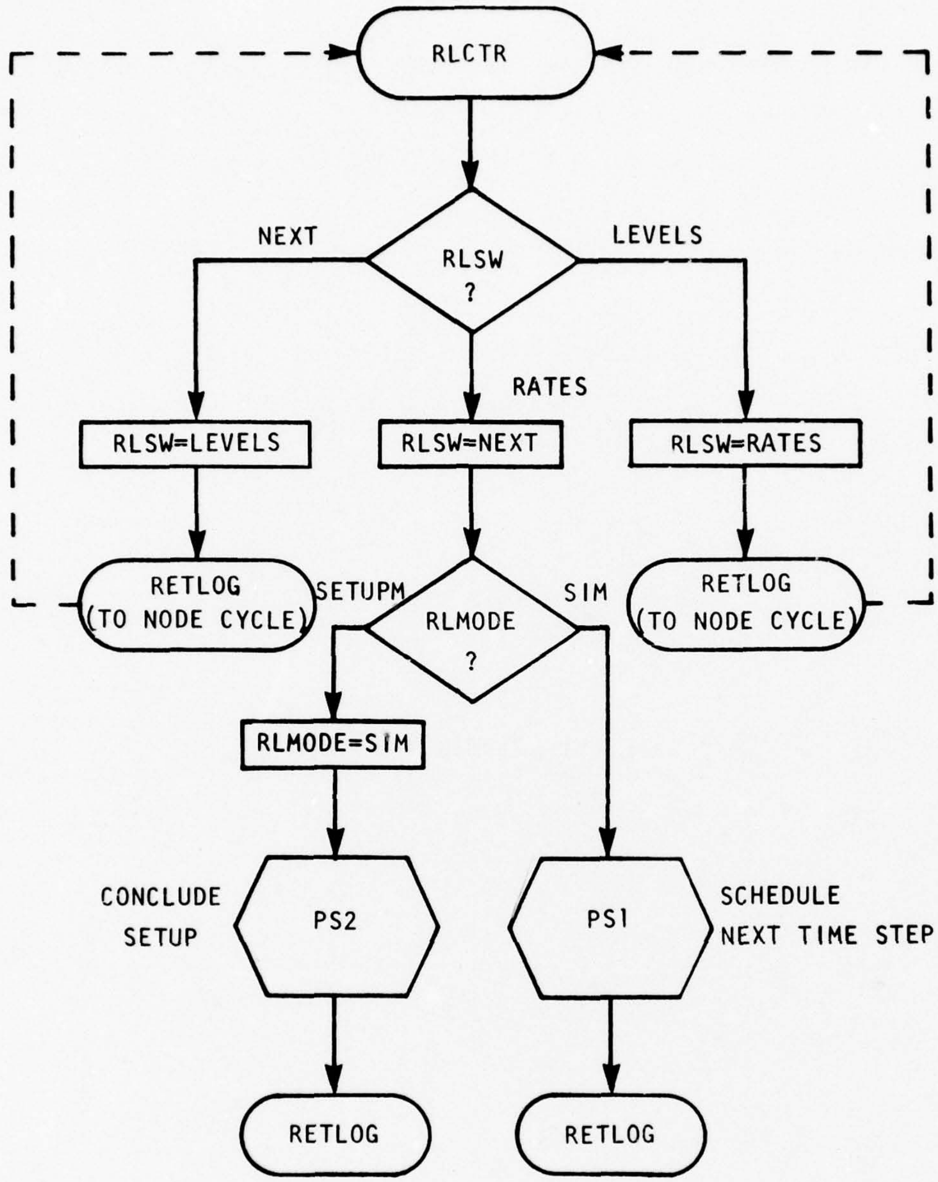
Verbs. None.

Other. None.

Input/Output Files Used

None.

(RLCTR-2)



RLCTR Control Logic

THIS PAGE INTENTIONALLY LEFT BLANK

RLDGN (RDSTYP, IEXOG, IXD, IXR)

General Description

This verb generates exogenous demand rates for rate/level rates of the type stored in the DTABLE stack pointed to from RDSTYP.IXD. The logic consists of two phases - a normal one and a setup one. In both cases IDSLOO is used to loop through all datasets of type RDSTYP. In the normal case, VDISTR is called to obtain a value from the random variable whose index is in RDSTYP.IEXOG, and this rate is stored in the first position of the rates stack at DTABLE (*,RDSTYP.IXD). In the setup phase, a rates stack is created in the pushdown stack array IZHOLD and the negative of its location is stored in RDSTYP.IXD. It is assumed that RLFLK or a similar module will be used to convert the setup stacks, etc. to DTABLE.

Assembler Inputs

Arguments.

RDSTYP - rate dataset type to loop on and generate exogenous demands

IEXOG - position in RDSTYP dataset that contains the index in PARAM array of the exogenous demand rate parameters

IXD - position in RDSTYP dataset that contains exogenous demand arrival rate stack pointer to DTABLE

IXR - not currently used

Parameter Slots. None.

Examples

RLDGN can be used to generate exogenous demands every time step.

```
EXDGN.RLDGN (P = *EXRAT, 1, 2, 0),
    *NXNOD
```

GASP Files Used

None.

Permanent Attributes Accessed

PDS Datasets:

(RLDGN-1)

All RDSTYP datasets are processed

RDSTYP.IEXOG - index to demand distribution parameters

RDSTYP.IXD - index to rate stack in DTABLE

Verb Inputs

From Calling Program.

Arguments as above

In /RLSYS/:

RLMODE - current mode model is operating in

SETUPM - flag for setup mode that RLMODE is compared to

From VDISTR.

In /ZMAWSY/:

ZSWT - random variable value

From RLLNK. None.

Verb Outputs

To VDISTR.

Argument -

First - index of parameters in PARAM array to use

To RLLNK.

Arguments -

First - node name

Second - location of rate stack pointer in LKDST dataset

Third - zero rate value

Fourth - no delay type

In /RLSYS/:

LKCDS - coordinates of LKDST dataset

To Permanent Attributes.

In /TBLCOM/:

DTABLE (2, pointer) - exogenous demand rate generated

To Calling Program. None.

Programs Called

(RLDGN-2)

Verbs. VDISTR.

Other. IDSLON, RLLNK.

Input/Output Files Used

None.

(RLDGN-3)

THIS PAGE LEFT INTENTIONALLY BLANK

RLDLK (NRLNOD, DLY)

General Description

Defines a rate/level demand link, using RLLNK, by establishing node NRLNOD as the next node in the demand rate stack pointed to from LKDST (LKCDS).IXDRAT. Thus, the link defined is from the current node, RLNOD, to NRLNOD. Delay DLY is associated with this link.

Assembler Inputs

Arguments.

NRLNOD - next node to be established in the demand rate stack - ending node for link currently being defined

DLY - delay to be associated with new demand link defined from RLNOD to NRLNOD

Parameter Slots. None.

Examples

RLDLK is the standard verb to define a demand link.

```
NODEA.
      ACSUP (2 = RLRLK (P = *NODEA, 3.),
                 RLDLK (P = *SUPTA, 1.))
```

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/:

RLMODE - control variable - equals SIM when simulation logic is to be executed; equals SETUPM when link definition or modification logic is to be executed

SETUPM - value used by RLMODE to signal set up mode

IXDRAT - index of element in LKDST (LKCDS) corresponding to demand flow paths of current interest

DRATE - demand flow rate associated with demand link of current interest

(RLDLK-1)

Verb Inputs

From Calling Program.

Arguments as specified above.

From RLLNK. None.

Verb Outputs

To RLLNK

Arguments -

NRLNOD - next node to be established in the demand rate stack

IXDRAT - index of element in dataset corresponding to demand flow paths of current interest

DRATE - demand flow rate of demand link of current interest

DLY - delay to be associated with new demand link being defined

Programs Called

Verbs. None.

Other. RLLNK.

Input/Output Files Used

NPRNT - Print File.

RLFLK General Description

Converts flow network descriptions from setup form to reference form. If the conversion is in connection with changes to the network, a final step of conversion is relegated to PS1 so that any transfer of flow rates and in-transit resources from old to revised portions of the network may be accomplished while parallel structures exist. A change report is printed by RLFLK. As is illustrated in the example below, RLFLK is intended for use at the conclusion of a setup pass, after all network definitions or changes for the current setup time have been recorded by RLLNK. Details of the data structures are given in Chapter III of this Part of the Module Catalog. Briefly, there are two stages to the conversion process. In the first, membership-tracing linkages are converted. In particular, pushdown stacks of pointers into RLC are transferred from IZHOLD to LCIX in /RLSYS/, and pushdown stacks of dataset types having coordinates in RLC are transferred to DTABLE in /TBLCOM/. The pointers in LDSC datasets are adjusted to point to their respective lists of dataset type names in DTABLE, and the pointers in the dataset type entries, which point to RLC entries, are adjusted to point into LCIX. In the second stage of the conversion, flow paths that are represented in pushdown stacks are transferred to DTABLE, and pointers are adjusted accordingly. If the stack representation includes changes to existing flow paths, a revised version of the path is built and stored in DTABLE. The old version of the path is retained in the stacks for use in PS1. At this stage an IZHOLD entry points (negatively) to the old path from IZKC and (positively) to the new path from IZK. The location of the pointer entry is stored in the PDS element (negatively) that in reference form points to the path in DTABLE. These activities are outlined in the accompanying figure.

When PS1 is executed, RLFLK assumes that all remaining conversion is completed there. Whether or not PS1 is executed, RLFLK returns control to the calling routine with a RETLOG.

Assembler Inputs

Arguments. None.

Parameter Slots.

1 - when there are changes to the flow network, logic to complete conversion of the network to reference form, to include reassignment of flow rates and resources in transit (vMAPLK is designed to perform this function)

Examples

vRLFLK was designed for use in the system dynamics control node in the following arrangement:

```
SDYNC.RLCTR (1 = TSTEP ( . . .)  $
             2 = RLSU (1 = RLFLK (1 = MAPLK)), FADE),
     *[next node in system dynamics node cycle]
```

Note that FADE is used to terminate the setup pass and return control to the simulation time advance and event sequencing logic.

GASP Files Used. None.

Permanent Attributes Accessed

PDS Dataset Elements.

LDSC (node).1 - pointer to dataset types that own flow paths of which node is a member

LKDST (LKCDS).[ptr] - elements that point to flow paths, for all dataset types (LKDST) and all sets of coordinates (LKCDS) that own flow paths

RLD(dly) - delay datasets associated with links that are being defined or changed

Common /LC/.

NFRELC - number of reclaimable entries in LCIX

Common /TBLCOM/.

NFREDT - number of reclaimable entries in DTABLE

Verb Inputs

From Calling Program. None.

From DTBGC (NDUM). DTABLE with maximum available space for new or revised portions of flow network data structure

From LCIXGC. LCIX with maximum available space for new or revised portions of flow network data structure

From IDSLON (JLDSC, LDSC, NEL).

Arguments:

JLDSC - if > 0, position in DSPOOL of current LDSC element
- if ≤ 0, signal that all LDSC datasets have been accessed

NEL - 1, number of elements in an LDSC dataset

In /DPTRS2/ - COORD (1), node number for current LDSC

From 1st ZREMEN (Statement 20). In /ZSTACK/ - entry from stack of dataset types for current LDSC:

IZREF -> 0, location in IZHOLD of next entry in stack, or 0 if this is last

IZKC - dataset type name

IZK - pointer to stack of RLC pointers for this dataset type

From 2nd ZREMEN (Statement 30). In /ZSTACK/ - entry from stack of RLC pointers for current dataset type for current LDSC

IZREF -> 0, location in IZHOLD of next entry in this stack, 0 if last

IZKC - packed ((number of times current LDSC node appears in paths owned by current dataset) X 10_8) + number of coordinates in RLC entry that, when applied to current dataset type, identifies current dataset)

IZK - index in RLC of current coordinates

From IXLCIX (ILCXSC). Function value, ILCX, location in LCIX of 1st available position for current RLC pointer list

From PERMDS (RELSET, LDSC, COORD(1)). None.

From INDTB (SCRTAB, IXSCR, INTBI) (Statement 42). Argument INTBI:

(RLFLK-3)

$\geq$ 0, index in DTABLE of location preceding 1st position of list of dataset types just stored; < 0, indicator that DTABLE contained inadequate space for the list

From IDSLON (JRDST, RATDST (IRATDT), NELR). Argument:

JRDST - index in DSPOOL of 1st element in current dataset of type RATDST (IRATDT)

NELR - number of elements in dataset pointed to by JRDST

in /DPTRS2/ - COORD, coordinates of dataset pointed to by JRDST

From PERMDS (GETSET, RATDST (IRATDT), COORD, 0, POOL). Argument:

POOL, elements of dataset just found by IDSLON

From ZDUPST (IZRFRN). Argument IZRFRN, index in IZHOLD of top of duplicate of stack representing current old flow path

From 3rd ZREMEN (After Statement 122). In /ZSTACK/ - entry in "new" flow path stack pointing to a two entry change stack

IZREF - >0, index in IZHOLD of next entry in new flow path

IZKC - CHCOD

IZK - pointer to two-entry change description stack

From 4th ZREMEN. In /ZSTACK/ - top entry of 2-entry change description stack

IZREF - index in IZHOLD of remaining entry in change stack

IZKC - old node number and old delay coordinate, packed

IZK - flow rate in old path

From ISTAKR (IZRFRN, IABS (IZKC)/IPAK, IPRED). Function value, IOLD, index in duplicate of old stack of entry whose node number is the same as the one just removed from change pair. Argument IPRED, index in IZHOLD of predecessor in flow path

From 5th ZREMEN (Statement 12201). In /ZSTACK/ - remaining entry in current change pair - to be discarded because of error

IZREF - 0

IZKC - "new" node and delay index of change

IZK - "new" flow rate

From ADJLC (ICTYP, INODO, RATDST (IRATDT)). Adjusted count (in LCIX entry) of references to INODO in flow paths owned by datasets of type RATDST (IRATDT) with current coordinates in COORD

From ADJLC (ICTYP, INODN, RATDST (IRATDT)). Adjusted count (in LCIX entry) of references to INODN in flow paths owned by dataset of type RATDST (IRATDT) with current coordinates in COORD

From PERMDS (GETSET, RLD, FLOAT (IDLYO), 0, RL). Argument - RL - elements of RLD delay dataset for "old" link of change. In /DSNOW/, NE - number of elements in current RLD dataset

From DLYLEV (RL, NE). Function value, DLEVO, amount of resource in delay just retrieved

From 6th ZREMEN. In /ZSTACK/ - remaining, "new", entry in change pair

IZREF - 0

IZKC - "new" node and delay index of change pair

From 7th ZREMEN. In /ZSTACK/ - entry deleted from duplicate flow path stack because change type is deletion

From ISTAKR (IZRFRO, INODO, IDUM). Function value, IRATEO, location of "old" node of change pair in orginal flow path, to permit retrieval of flow rate from RATATT stack for that link and saving it in its normal position as 2nd element of IRATEO entry

From ZRELST. None. (RATATT stack released).

From ADJLC (ICTYP, INODN, RATDST (IRATDT)) (Before Statement 12502). In RLC pointer entry in LCIX, adjusted count of references to node INODN from flow paths owned by dataset of type RATDST (IRATDT) with current coordinates in COORD.

From 1st and 2nd ZPLAENs. (Around Statement 126). In /ZSTACK/ - location of entry just added to duplicate flow path, i.e., path being revised

From ADJLC (ICTYP, INODN, RATDST (IRATDT)). In RLC pointer entry in LCIX, adjusted count of references to INODN in flow paths owned by dataset [RATDST (IRATDT)] (COORD)

(RLFLK-5)

From 8th ZREMEN (After Statement 131). In /ZSTACK/ - entry removed from revised flow path stack for transfer to DTABLE

IZREF - >0, index in IZHOLD of next entry in path; 0, signals current is last entry

IZKC - packed node number and delay index

IZK - initial rate of flow

From PERMDS (VALEL, RLD, FLOAT (IDLY), I, RL(1)). Argument RL(1), index of delay parameter and form for current link which was in old path

From ARLDIX (DLY, 0). Argument DLY, coordinate for RLD dataset for link in new path

From INDTB (SCRTAB, NELRN, INDTI). Argument INDTI, ≥ 0, index in DTABLE of first entry of segment in which new path was stored; < 0, signal that there wasn't enough space in DTABLE .

From 9th ZREMEN. In /ZSTACK/ - top entry of RATATT stack

IZREF - index in IZHOLD of remaining RATATT stack entry

IZKC - old flow rate for link owning RATATT

IZK - number of elements in this RATATT dataset

From 10th ZREMEN. In /ZSTACK/ - bottom entry of RATATT stack

IZREF - 0

IZKC - RATATT.1, value of first element

IZK - RATATT.2, if needed

From PERMDS (ADDSET, RATATT, FLOAT (IDT), NEL, RAT). None.

From PERMDS (SETSET, RATDST (IRATDT), COORD, NELR, POOL). None.

From ZERRPT. None.

Verb Outputs

To Permanent Attributes.

PDS Dataset Elements.

LDSC (node).1 - pointers to dataset type lists revised to point to DTABLE instead of pushdown stack, for all nodes in stacks that were defined or modified

(RLFLK-6)

LKDST (LKCDS).[ptr] - elements that point to flow paths that were defined, converted from pointer to pushdown stack to pointer to DTABLE

Common /LC/.

NFRELC - number of freed entries in LCIX possibly reduced to zero by garbage collection

NRLC - number of LCIX entries occupied possibly reduced by garbage collect but possibly increased by new links or further reduced by deletion of links

LCIX - occupancy pattern altered by effects noted above

Common /TBLCOM/.

NFREDT - number of freed entries in DTABLE possibly reduced by garbage collection

INDTBL - index of next available entry in DTABLE possibly modified by garbage collection, additions and deletions of flow links

DTABLE - occupancy pattern altered by effects noted

To Calling Program. None.

To DTBGC (NDUM). None.

To LCIXGC. None.

To IDSLON (JLDSC, LDSC, NEL). Argument LDSC. Also, in /PDSRET/, IPDRET, address for return from current IDSLON

To 1st ZREMEN. In /ZSTACK/ - IZREF location in IZHOLD of next entry in stack of dataset types, for current LDSC, to be removed to ISCRTB for subsequent transfer to DTABLE

To 2nd ZREMEN. In /ZSTACK/ - IZREF, location in IZHOLD of next entry in stack of pointers to RLC, for current dataset type, to be removed to LCIX via scratch array LCXSCR

To IXLCIX (ILCXSC). Argument ILCXSC, number of locations required in LCIX for current list of RLC pointers in LCXSCR

To PERMDS (RELSET, LDSC, COORD(1)). All arguments, including COORD(1), the current LDSC node number, so that current LDSC can be released -

(RLFLK-7)

node is no longer in any flow paths

To INDTB (SCRTAB, IXSCR, INTBI). Arguments SCRTAB, list of dataset types for current LDSC, and IXSCR, number of dataset types in list

To IDSLON (JRDST, RATDST (IRATDT), NELR). Argument RATDST (IRATDT), path-owner dataset type currently being processed; also IPDRET in /PDSRET/ as IDSLON return address

To PERMDS (GETSET, RATDST (IRATDT), COORD, 0, POOL). All arguments.

To ZDUPST (IZRFRN). In /ZSTACK/ - IZREF, index in IZHOLD of top entry of stack to be duplicated, namely stack representing current flow path for which changes have been specified

To 3rd ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of next entry in stack of change pointers

To 4th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of top of change pair pointed to by change pointer retrieved by 3rd ZREMEN

To ISTAKR (IZRFRN, IABS (IZKC)/IPAK, IPRED). Arguments IZRFRN, index in IZHOLD of stack into which revisions are being introduced, and IABS(IZKC)/IPAK, the node number

To 5th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of second entry in change pair pointed from 3rd ZREMEN, when change cannot be carried out

To ADJLC (ICTYP, INODO, RATDST (IRATDT)). All arguments. ICTYP is positive change code, INODO is node whose path membership counter for current owner dataset is to be adjusted because specified change cannot be implemented

To ADJLC (ICTYP, INODN, RATDST (IRATDT)). Same as above for changes of types other than deletion; this is a second adjustment of the counter for substitution type changes

To PERMDS (GETSET, RLD, FLOAT (IDLYO), 0, RL). All arguments except RL.

To DLYLEV (RL, NE). Arguments RL, elements of delay whose resource content is to be determined, and NE, number of elements in RL

To 6th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of second entry in change pair pointed to from 3rd ZREMEN, when change can be carried out

(RLFLK-8)

To 7th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of flow path entry to be deleted from stack being revised

To ISTAKR (IZRFRO, INODO, IDUM). Arguments IZRFRO, top of "old" flow path stack, and INODO, node in that path whose link had a RATATT dataset

To ZRELST. In /ZSTACK/ - IZREF, index in IZHOLD of RATATT stack to be released because link was deleted

To ADJLC (ICTYP, INODN, RATDST (IRATDT)). Arguments. ICTYP is positive change code; INODN is node number specified for insertion but found to be already present in path

To 1st ZPLAEN. In /ZSTACK/ - IZREF, location in IZHOLD to which a link is to be added (pushdown) to a flow path

To 2nd ZPLAEN. Same as 1st, but for addition before a specified current node instead of after (procedure is still pushdown)

To ADJLC (ICTYP, INODN, RATDST (IRATDT)). Arguments. ICTYP is positive change code, INODN is node specified as replacement for an existing node but found to be already present

To 8th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of next entry to be removed from current flow path stack for transfer to DTABLE via SCRTAB

To PERMDS (VALEL, RLD, FLOAT (IDLY), 1, RL(1)). Arguments.

To ARLDIX (DLY, 0). Arguments. DLY is delay type for which an RLD dataset is to be generated for current link

To INDTB (SCRTAB, NELRN, INDT1). Arguments SCRTAB, new or revised flow path in DTABLE format, and NELRN, number of entries in flow path

To 9th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of top of RATATT stack whose RATATT data are to be assigned to current link

To 10th ZREMEN. In /ZSTACK/ - IZREF, index in IZHOLD of second entry in RATATT stack to be assigned to current link

To PERMDS (ADDSET, RATATT, FLOAT (IDT), NEL, RAT). Arguments. IDT is location in DTABLE of link to which current RATATT is to be assigned; NEL is length of this dataset; RAT are the elements

To PERMDS (SETSET, RATDST (IRATDT), COORD, NELR, POOL). Arguments.

POOL contains path-owner pointers for new paths and unchanged previous paths pointing to DTABLE, pointers for changed paths still pointing to IZHOLD for further processing and final conversion in PS1

To PS1.

Flow network in reference form except paths that were modified. For them the old path remains in IZHOLD, the new path is in DTABLE, and the owner-pointer points to an IZHOLD entry whose IZKC points to old and IZK to new

To ZERRPT. None.

Program Called

Verbs. None.

Other. ADDSET, ADJLC, ARLDIX, DLYLEV, DTBGC, GETSET, IDSLON, INDIB, IPAK2, IVNPK2, ISTAKR, IXLCIX, LCIXGC, LIN, PERMDS, PGHDR, SETSET, ZDUPST, ZRELST, ZSTAK.

Input/Output Files Used

NPRNT - print file, for flow network change report

(RLFLK-10)

DTABLE garbage collection LCIX garbage collection	to minimize risk of incurring during conversion

LOOP (use IDSLON) to find all LDSC datasets. For each:

> If LDSC.1 < 0 transfer LKDSTs from IZHOLD to DTABLE and associated pointers to RLC from IZHOLD to LCIX. Omit LKDSTs for which there are no RLC pointers. Delete the LDSC dataset if there are no active LKDSTs.

LOOP through rate dataset types in table RATDST of all types for current model. For each type:

> LOOP (use IDSLON) through all datasets of current type. For each:
>
> > Check pointer elements. For any that are negative:
> >
> > 1. If there are no revisions to a previous version of the rate stack, transfer it to DTABLE; assign RL delays when a link reqoires it,
> > 2. If there are revisions, create a duplicate of the original version. Then read changes from "new" stack and its associated change pairs and introduce them to the duplicate of the original stack. Finally, transfer the revised stack to DTABLE as in 1, assigning delays and reestablishing RATATTs as appropriate.

If there were no changes, Retlog.

If there were changes, execute PS1 to map old rates and delay content to new. Then Retlog.

Outline of vRLFLK Logic
(RLFLK-11)

THIS PAGE LEFT INTENTIONALLY BLANK

RLMSU

General Description

Sets RLMODE, a control variable in /RLSYS/, to SETUPM, the value used to signal setup mode in the model when rate/level links are to be defined or changed.

Assembler Inputs

Arguments. None.

Parameter Slots. None.

Examples

RLMSU can be used to initiate a setup pass in a model after changes have been made.

MODMP (P = 5), CHPDS (P = 5), CHRLD (P = 5), RLMSU, * RLCTR

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

RLMODE - control variable - equals SIM when simulation logic is to be executed; equals SETUPM when link definition or modification logic is to be executed

SETUPM - value used by RLMODE to signal set-up mode

Verb Inputs

From Calling Program. None.

Verb Outputs

To Permanent Attributes.

In /RLSYS/:

RLMODE - control variable, has been set to SETUPM to indicate setup mode

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used. None.

(RLMSU-1)

THIS PAGE LEFT INTENTIONALLY BLANK

RLNOD (RLNAM, RLN)

General Description

This verb is used to assign a rate/level node name and number. It sets RLNOD and KRLNOD to RLN. In setup pass, establishes RLNNAM in /RLSYS/ as table of RL node names. In doing this, the name RLNAM is stored in position RLNNAM (RLN). If previous content was nonzero and non-RLNAM an error message is printed and the old name is overwritten.

Assembler Inputs

Arguments.

RLNAM - RL name to be stored in table RLNNAM

RLN - number of node name RLNAM

Parameter Slots.

None.

Examples

The RLNOD verb must be used at the beginning of every rate/level node to set the node number. If a subnode contains rate/level verbs and it is entered from another node, RLNOD must be used again to set the node number.

```
NODEA. RLNOD (P = *NODEA, 3.),
              .
              .
              .
          *NODEB.2
NODEB.
      /2/ RLNOD (P = *NODEB, 4.),
```

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RLNOD - current RL node number (floating point)

KRLNOD - current RL node number (integer)

(RLNOD-1)

RLMODE - control variable

SETUPM - value used by RLMODE to signal set up mode

RLNNAM - names of system dynamics nodes

NRLN - number of system dynamics nodes is current model

NRLNMX - maximum number of system dynamics nodes current model can accomodate

Verb Inputs

From Calling Program.

Arguments - see above

Verb Outputs

To Permanent Attributes

In/RLSYS/:

RLNOD - node number RLN

KRLNOD - node number RLN (integer)

RLNNAM(I) - RLNAM, name of I-th system dynamics node, I=RLN

NRLN - number of system dynamics nodes in current model

Programs Called

Verbs. None

Other. None

Input/Output Files Used

NPRNT - Print File

Pages IV- 91 and 92

Not available

RLPDS

General Description

Basic input verb for Rate/Level supply logic, defines PDS tables and dataset types and reads datasets. Any variable in a dataset may be flagged for statistics collection. The format for RLPDS input data cards is shown in the accompanying tables.

At the end of an input deck, PDS table descriptors and DSPOOL occupancy is printed out. Statistics collection is organized by frequency group by a call to routine RLSSET.

Assembler Inputs

Arguments. None.

Parameter Slots. None.

Examples

RLPDS is generally included in the input node ZINIT.

ZINIT. RLPDS, IPDS, . . .

GASP Files Used. None.

Permanent Attributes Accessed. None.

Verb Inputs

From Calling Program. None.

Verb Outputs

To RLINT.

Arguments:

DSETNC - dataset name

VELC - value of element from the input card.

To DEFTAB.

In /BUF/:

BFF - array of PDS table names and dimensions

To DEFDST.

In /BUF/:

BFF - array of PDS dataset type names and element names

To Calling Program. None.

Programs Called

Verbs. None.

Other. CLEAR, CLERI, DEFDSI, DEFDST, DEFTAB, INEROR, PRTAB, RLINT, RLSASN, RLSSET, RLSTAT.

Input/Output Files Used

NPRNT - Print File

NCRDR - Card Read File

TABLE RLPDS INITIAL CARD FORMAT

CARD COLUMNS	FORTRAN FORMAT	DESCRIPTION
1-5	A5	"RLPDS", CARD NAME
6	A1	"1", CARD TYPE
7-10	4X	BLANK
11-20	F10.3	BASIC STATISTICS COLLECTION INTERVAL IN NUMBER OF TIME UNITS

(RLPDS-3)

TABLE RLPDS COORDINATE CARD FORMAT

CARD COLUMNS	FORTRAN FORMAT	DESCRIPTION
1-5	A5	"RLPDS", CARD NAME
6	A1	"2", CARD TYPE
7	1X	BLANK
8-13	A6	DATASET NAME
14	1X	BLANK
15-20	A6	DATASET IDENTIFIER FOR CARD
21-30	F10.0	FIRST COORDINATE VALUE
31-40	F10.0	SECOND COORDINATE VALUE[a]
41-50	F10.0	THIRD COORDINATE VALUE[a]
51-60	F10.0	FOURTH COORDINATE VALUE[a]
61-70	F10.0	FIFTH COORDINATE VALUE[a]
71	1X	BLANK
72-73	I2	NUMBER OF ELEMENTS IN DATASET

[a]SECOND THRU FIFTH COORDINATES ENTERED ONLY AS NECESSARY.

(RLPDS-4)

TABLE RLPDS ELEMENT VALUES CARD FORMAT

CARD COLUMNS	FORTRAN FORMAT	DESCRIPTION
1-5	A5	"RLPDS", CARD NAME
6	A1	"3", CARD TYPE
7	1X	BLANK
8-13	A6	DATASET NAME
14	1X	BLANK
15-20	A6	DATASET IDENTIFIER FOR CARD, MUST MATCH ID ON RLPDS2 CARD
21-28	F8.0	VALUE OF FIRST ELEMENT OF DATASET
29-30	I2	STATISTICS COLLECTION FLAG FOR FIRST ELEMENT
31-38	F8.0	VALUE OF SECOND ELEMENT
39-40	I2	STATISTICS COLLECTION FLAG FOR SECOND ELEMENT
.	.	.
.	.	.
.	.	.
71-78	F8.0	VALUE OF SIXTH ELEMENT
79-80	I2	STATISTICS COLLECTION FLAG FOR SIXTH ELEMENT

ADDITIONAL RLPDS3 CARDS ARE REQUIRED IF THE DATASET HAS MORE THAN 6, 12, OR 18 ELEMENTS.

(RLPDS-5)

THIS PAGE LEFT INTENTIONALLY BLANK

RLRLK (NRLNOD, DLY)

General Description

Defines a rate/level return link, using RLLNK, by establishing node NRLNOD as the next one in the return rate stack pointed to from LKDST (LKCDS).IXRRAT. Thus the link defined is from NRLNOD to the one below it in the return rate stack. Delay DLY is associated with the link.

Assembler Inputs

Arguments.

NRLNOD - next node to be established in the return rate stack

DLY - delay associated with new return link defined from NRLNOD to the node below NRLNOD in the return rate stack

Parameter Slots. None.

Examples

RLRLK is the standard verb to define return links to a node. BRLKR can be used to set up multiple return paths.

```
NODEA.
    ACSUP (2 = BRLKR (P = 2 $
                        1 = RLRLK (P = *NODEA,2.)$ + R1 PATH
                        2 = RLRLK (P = *NODEA, 4.))) + R2 PATH
```

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/:

RLMODE - control variable

SETUPM - value used by RLMODE to signal set up mode

IXRRAT - index of element in LKDST(LKCDS) corresponding to return flow path of current interest

RRATE - return flow rate associated with return link of current interest

(RLRLK-1)

Verb Inputs

From Calling Program.

Arguments as specified above.

To RLLNK. None.

Verb Outputs

To RLLNK.

Arguments:

NRLNOD - next node to be established in the return rate stack

IXRRAT - index of element in dataset corresponding to return flow path of current interest

RRATE - return flow rate

DLY - delay associated with new return link

Programs Called

Verbs. None.

Other. RLLNK.

Input/Output Files Used

NPRNT - Print File.

(RLRLK-2)

RLSU

General Description

Rate/level setup module, merely calls PS1 currently, intended as a mnemonic device for use in PS2 of RLCTR.

Assembler Inputs

Arguments. None.

Parameter Slots.

1 - logic to convert setup data to permanent (e.g., RLFLK)

Examples

RLSU is merely a mnemonic place holder to insure that a rate/level setup step is done to convert the initial form of the flow data structure to the permanent reference form in PS2 of RLCTR.

```
RLCTR.  RLCTR (1 = TSEP (1 = DELAY (P = 0), * RLCTR)  $
                   2 = RLSU (1 = RLFLK),  + SETUP
                       FADE )    + RETURN CONTROL AFTER SETUP
```

GASP Files Used. None.

Permanent Attributes Accessed. None.

Verb Inputs

From Calling Program. None.

Verb Outputs

To Permanent Attributes. None.

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

NPRNT - Print File

THIS PAGE LEFT INTENTIONALLY BLANK

RSTAK (RPTRS)

General Description

This verb prints all rate/level rate stacks of type RPTRS. The report produced is invaluable in verifying the structure of demand and receipt paths and in determining the flows in the model. This is the Flows in Rate/ Level Links report shown in the attached figure. The report shows for each dataset at a node, each path and the current rate of flow and delay content on each link in the path. As an example, consider the node named 67IN in Figure III-1. For class 902, the node has two demand paths and two receipt paths defined. The primary demand path has a link into node 67IN with a rate of 8307 pounds, a link into node 9GS with a rate 0, and a link into node CONUS with a rate of 7251 pounds. The secondary demand path is inactive but links the node 67IN to the node WRSL. The primary receipt path has a "bottom" link into the CCPE node with a rate of 4893 pounds over the time step. This link has a total of 34,252 pounds delayed for a period of 7 days. The second link into the node 67IN has a flow of 2,526 pounds with 247,034 pounds delayed over a 46.5 day period. This class of materiel has no secondary receipt path, indicated by the absence of an FCRAT.6 pointer, but has a tertiary receipt path into node 67IN which is inactive. This report should be checked during the initial model runs to verify that each node description and dataset is correct. In later runs, the report will be used to analyze flows over the links in the model.

Assembler Inputs

Arguments.

RPTRS - alpha name of rate stack dataset type, currently EXRAT or FCRAT are the only types recognized

Parameter Slots. None.

Examples

RSTAK should be included in the report node ZRPRT in a continuous flow model.

ZRPRT. ENDTN, RPSTG, CSRPT, CONRP, RSTAK (P = * EXRAT), RSTAK
(P = * FCRAT), . . .

GASP Files Used. None.

Permanent Attributes Accessed

PDS Datasets:

All datasets of type RPTRS

RLD - rate/level delay datasets for links in model

In /RLSYS/:

RLNNAM - array of rate/level node names

RLDPAR - rate/lvel delay parameters

In /TBLCOM/:

DTABLE - rate stacks pointed to from RPTRS datasets

Verb Inputs

From Calling Program.

Argument - RPTRS dataset type name.

From IDSLON.

Arguments -

IR - index to first word of RPTRS dataset in DSPOOL

NEL - number of elements in RPTRS dataset

From DLYLEV.

Function Value - total content in RLD delay cells

Verb Outputs

To IDSLON.

Arguments -

RPTRS - name of PDS dataset to loop on

To DLYLEV.

Arguments -

RL - array containing elements of RLD dataset

NE - number of elements in RLD dataset

To Calling Program. None.

Programs Called

Verbs. None.

(RSTAK-2)

Other. DLYLEV, HASHI, IDSLON, IUNPAK, LIN, MSHIFT, PERMDS.

Input/Ouput Files Used

NPRNT - output file for report printed

(RSTAK-3)

AD-A040 804 MODELS OF THE US ARMY WORLDWIDE LOGISTIC SYSTEM (MAWLOGS) VOLUME III MODU..(U) BDM CORP VIENNA VA FEB 77 BDM/W-76-211-TR-VOL-3-PT-6 DAAG39-76-C-0134

3/4

UNCLASSIFIED F/G 15/5 NL

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Blank

SVRL

General Description

SVRL is a restart save module which saves, on a file, current status of system dynamics portions of model for later use in restart. The file layout is described in the table below.

Record Number	Description
1	"SVRL", the data identifier
2	TSEP, RLDST, TNXT, RLNNAM, NRLN, NRLNMX, RL, RLDPAR, NRLDPR, RLDNAM, DLRIXK, RLDTYP, RATDST, NRATDT, i.e., variables in /RLSYS/
3	KRLST, KRLST1, NRLST, NXRLST, RLSTIU, RLSTIS, i.e., variables in /RLSYS/
4	NRLDIM, NRLCSV, NLCSV, NXTRLC, NXTLCX, NFRELC, i.e., variables in /LC/
5	RLC - set of PDS coordinates for datasets that represent or own system dynamics flow paths
6	LCIX - array of pointers into RLC
7	RLSNAM in COMMON/STNAM/

Assembler Inputs

Arguments. None

Parameter Slots. None

Examples

SVRL should be included in subnode 1 of the node ZREST to save rate/level variables.

```
ZREST.  SVRUN (1 = SVPDS, SVRL, SVSTA)
  /2/   IZRUN (1 = IZPDS, IZRL, IZSTA)
```

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/: see record numbers 2 and 3 of table above.

In/LC/: all variables except, NRLC, NLCIX, LRLCHG, and LRLPTR

In/STNAM/: RLSNAM

Verb Inputs

From Calling Program.

None.

Verb Outputs

To Calling Program.

None.

To Permanent Attributes.

None.

To Restart File.

All permanent attributes accessed

Programs Called

Verbs. None

Other. None

Input/Output Files Used

NFREST - Restart File

NPRNT - Print File

TSTEP

General Description

Rate/level time step module increments time step counter and schedules next time step through time file.

Assembler Inputs

Arguments. None.

Parameter Slots.

1 - schedule next time step (D - delay)

Examples

TSTEP is used in continuous flow models to advance the time step. It is designed for PS1 of verb RLCTR in the rate/level control node of a model.

```
RLCTR.RLCTR (1 = TSTEP (1 = DELAY (P = 0), *RLCTR) $
                        + SCHEDULE NEXT TIME STEP
             2 = RLSU (1 = RLFLK),   + SETUP LOGIC
                 FADE ),
       *EXDGN    $
```

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/:

TSTEP - time step, in number of simulation time units

NSTEP - number of time steps executed to date

In /VRBGSP/:

ATRIB(IM) - array for transmitting attributes of a temporary entity to and from GASP files and among model verbs and routines

TNOW - current simulation time

IATT - number of attributes available for logistics data. IM-IATT are reserved for system data

(TSTEP-1)

Verb Inputs

From Calling Program. None.

Verb Outputs

To Permanent Attributes.

In /RLSYS/:

NSTEP - number of time steps executed to date

In /VRBGSP/:

ATRIB - array for transmitting attributes of a temporary entity to and from GASP files and among model verbs and routines. First attribute used here to schedule next time step

Programs Called

Verbs. None.

Other. CLEAR.

Input/Output Files Used

NPRNT - Print File.

CHAPTER V

CONTINUOUS SERVICE ROUTINES

THIS PAGE LEFT INTENTIONALLY BLANK

ABOXST (DSTN, CDS, XINCR)

General Description

Updates boxcar train for observing time delays in a rate/level model by adding XINCR to the boxcar (BOX) currently being filled. The boxcar data are assumed stored in PDS dataset DSTN (CDS) in the following form -

Element	Name	Definition
1	IOLD,NMAX	Index in X of oldest box, and dimension of X
2	INEW,NCUR	Index in X of newest box, and number of boxes in use
3	XW	Width of a box, in time steps
4	TOLD	Oldest time in current content of IOLD
5	FOLD	Fraction of oldest box currently occupied
6	X(4)	
7	X(5)	
8	X(6)	Assumes IOLD = 5, INEW = 4, NMAX = NCUR = 7
9	X(7)	
10	X(1)	X(K) is the amount whose current age is about
11	X(2)	TNOW-(XW*TSTEP*(NMAX-K))
12	X(3)	

If all cells are full, COMPRB is called to compress the cells.

Arguments.

DSTN - dataset type in which boxcar data are stored

CDS - coordinates of boxcar data in DSTN

XINCR - resources added to boxcar cell currently being considered

Examples

ABOXST is called when demands are sent out such as in CDMDS to add the due-in level to the OST measuring boxcar trains.

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

TSTEP - time step

In/DSNOW/:

IXDST - index of dataset type name and associated pointers

IXEL1 - index in DSPOOL of first element in dataset of current interest

In Dataset DSTN (CDS): All elements as described under general description.

Program Inputs

From Calling Program.

Arguments:

See above

From COMPRB

Arguments:

INC - Number of cells empty after compression

X, XW, and FOLD - revised values, see under Program Outputs

Program Outputs

To Permanent Attributes.

PDS Datasets: dataset type DSTN with coordinates CDS, incremented value in NCUR cell

To COMPRB.

Arguments:

X - Array of boxcar cells

NMAX - Number of cells in X

XW - Current width of a cell (in time steps)

IOLD - Index in X of current oldest cell

TOLD - Time of entry associated with oldest content of oldest cell

FOLD - Fraction of oldest cell currently occupied

To Calling Program.

None.

(ABOXST-2)

Programs Called

Verbs. None

Other. COMPRB, GETSET, IPAKIT, IUNPAK, PERMDS, SETSET

Input/Output Files Used

None.

THIS PAGE LEFT INTENTIONALLY BLANK

ADJLC
R/SS/CS
4/75

ADJLC (ICTYP, INOD, IRATDT)

General Description

Adjusts counter in LCIX of number of times a particular rate dataset of type IRATDT is linked to node INOD (RLNODE number). Adjustments are as follows -

ICTYP	CHANGE TYPE	ADJUSTMENT
1	DELETE	OLDNOD, +1
2 or 4	ADD	NEWNOD, -1
3	SUBSTITUTE	NEWNOD, -1

Adjustments are necessary when verb RLFLK detects that the change cannot be made or would duplicate an existing node in a path. When the adjustment causes the LCIX counter to become zero, the entry is deleted from LCIX. When the LCIX entry to be incremented does not exist and the increment is positive, it is created. The coordinates of the dataset of interest are in COORD in /DPTRS/. /DPTRS/ - interface between MAKPTR and HASHK and HASHL, and between PRTAB and COOR.

Arguments.

ICTYP - change type of 1, 2, 3, or 4

INOD - node that rate dataset of type IRATDT is linked to through LDSC

IRATDT - type of rate dataset

GASP Files Used. None.

Permanent Attributes Accessed

In /LC/:

LCIX - array of pointers into RLC

In /RLSYS/:

RLDST(6) - 6th dataset type name

In /TBLCOM/:

DTABLE - array for storing pairs of values, in this case rate stacks

(ADJLC-1)

In /DPTRS/:

COORD - coordinates array for IRATDT dataset

Program Inputs

From Calling Program.

Arguments - see above

From IRLC

Function Value - index of coordinates of IRATDT dataset in RLC

From IXDTI.

Function Value - index of rate dataset type in list of linked dataset types for INOD in DTABLE

From IXLCIX

Function Value - index of first word of space assigned in LCIX array

From NDIDST.

Function Value - number of dimensions in PDS dataset IRATDT

From RELCIX. None.

Program Ouputs

To IRLC.

Arguments -

COORD - coordinate values to search RLC for

NCDS - number of coordinate values in array COORD

To IXDTI.

Arguments -

IDTPTR - index to dataset type pointer list in DTABLE

IRATDT - rate dataset type as input to ADJLC

To IXLCIX.

Argument -

N - number of entries to be assigned in LCIX array

To NDIDST.

Argument -

IRATDT - rate dataset type as input to determine the number of dimensions (coordinates)

(ADJLC-2)

To RELCIX.

Arguments -

N - number of entries to be released from LCIX

I1 - index to first entry of LCIX to release

To Permanent Attributes.

In /LC/:

LCIX - array of pointers in RLC adjusted as described above

In /TBLCOM/:

DTABLE - array for string pairs of values

To Calling Program. None.

Programs Called

Verbs. None.

Other. IPAKZ, IRLC, IUNPKZ, IXDTI, IXLCIX, NDIDST, PERMDS, RELCIX.

Input/Output Files Used. None.

THIS PAGE LEFT INTENTIONALLY BLANK

ARLDIX (DLRIX, RATEIN)

General Description

Assigns an index and creates and RLD dataset for the delay whose form is represented by the negative index DLRIX. The returning value of DLRIX is the unique index taken from DLRIXK in /RLSYS/ and represents a specific delay of a specific rate. Note - RATEIN is furnished for potential use here to initialize the intermediate rates and levels.

Arguments.

DLRIX - negative index that represents the delay created in the RLD dataset

RATEIN - variable for potential use to initialize the intermediate rates and levels

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

RLD - RLDST(12) 12-th entry in list of PDS dataset type names

RL - array of elements of system dynamics delays

DLRIXK - last index assigned as the coordinate of an RL delay

RLDPAR (*, J) parameters for Jth system dynamics delay, J = 1, 2, . . ., NRLDPR, as follows:

RLDPAR (1, J) - code indicating form of delay

RLDPAR (2, J) - mean delay time for 1st and 3rd order delay; pointer to DTABLE for boxcar delays

Program Inputs

From Calling Program

Arguments - see above

Program Outputs

To Permanent Attributes.

In /RLSYS/:

RL(1) - RLDPAR index is set into element 1 of RL

DLRIXK - last index assigned as the coordinates of an RL delay - incremented by 1

To Calling Program.

Argument -

DLRIX - index assigned as the coordinate of the new RL delay

Programs Called

Verbs. None.

Other. ADDSET, CLEAR, PERMDS.

Input/Output Files Used. None.

(ARLDIX-2)

BLKRL

General Description

This routine is a block data program for rate/level service data structure. It initializes a relatively small number of variables in COMMON/RLSYS/ and /LC/. Certain dimensioning variables such as NLCIX and NRLC are initialized to values defined in the Model Assembler Dimensions Module. This is done via common module RLSIZ which is read by the Model Assembler and incorporated into BLKRL by a *CALL RLSIZ.

Arguments.

None.

GASP Files Used

None.

Permanent Attributes Accessed

None.

Program Inputs

From Calling Program.

None.

Program Outputs

To Permanent Attributes.

In COMMON/LC/

NXTRLC - index of next available position in RLC

NXTLCX - index of next available position in LCIX

In COMMON/RLSYS/

RLSW - control variable, equals NEXT, control is to be given to activities in which neither rates nor levels are to be updated.

LEVELS - value used by RLSW to signal levels update

RATES - value used by RLSW to signal rates update

RLMODE - control variables, equals SETUPM, link definition or modification logic to be executed

SETUPM - value used by RLMODE to signal setup mode

SIM - value used by RLMODE to signal simulation mode

RLDST(I) - I-th entry in list of PDS dataset type names containing elements potentially subject to system dynamics statistics collection

NXRLST - last index in KRLST, etc. assigned

NRLN - number of system dynamics nodes in current model

DLRIXK - last index assigned as the coordinate of an RL delay

<u>To Calling Program</u>.

None.

<u>Programs Called</u>

<u>Verbs</u>. None

<u>Other</u>. None

<u>Input/Output Files Used</u>

None.

CHRLST (JRLST, NMOVE, NEXT, NCRLST)

General Description

Change rate/level statistics collection, adjust pointers by NMOVE, and assign statistics not yet assigned.

JRLST Conventions -

-1 - turn statistic off

0 - no change

Pos - change frequency and type to new value

Also update RLSTAT and change counter NCRLST.

Arguments.

JRLST - array of indicators which direct CHRLST

NMOVE - number of places to adjust pointers

NEXT - next element in pointer array

NCRLST - counter for the number of changes made to the Rate/Level statistics collection

Examples

CHRLST is called by CHPDS to change the statistics collection patterns as specified.

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/

KRLST(I) partial identifier (see KRLST1) of Kth system dynamics statistical variable - contains (RLSTAT collection code *10000) + K, K = 1, 2, ..., NXRLST

KRLST1(K) remainder (after KRLST(I)) of identification of Kth system dynamics statistical variable - contains (STAT1 index *100000) + index in DSPOOL of value to be observed

In/DSNOW/

NE - number of elements in dataset of current interest

IXEL1 - index in DSPOOL of first element in dataset of current interest

Program Inputs

From Calling Program.

Arguments - JRLST, NMOVE, NEXT - see above

Program Outputs

To Permanent Attributes.

In/RLSYS/

KRLST(I) partial identifier (see KRLSTI) of Kth system dynamics statistical variable - contains (RLSTAT collection code *10000) + K, K = 1, 2, ..., NXRLST

KRLSTI(K) remainder (after KRLST(I)) of identification of Kth system dynamics statistical variable - contains (STATI index * 100000) + index in DSPOOL of value to be observed

To Calling Program.

Argument NCRLST - number of changes made to rate/level statistics

Programs Called

Verbs. None

Other. RLSASN

Input/Output Files Used

None.

COMPRB (X, NX, XW, IOLD, TOLD, FOLD, INC)

General Description

Compresses content of filled boxcar train of NX cells, each of width XW, into NX-1 or fewer cells of width XWNEW. XW and XWNEW are in integer time steps. Revised values for X, XW, and FOLD are returned. COMPRB is necessary when there are insufficient cells to hold the materiel for the delay time. This occurs when materiel is not removed in sufficient time form cell IOLD to make it available for new materiel being added. Compression makes the cells of larger time width and therefore able to hold more materiel.

Arguments.

X - Array of boxcar cells
NX - Number of cells in X
XW - Current width of a cell (in time steps)
IOLD - Index in X of current oldest cell
TOLD - Time of entry associated with oldest content of oldest cell
FOLD - Fraction of oldest cell currently occupied
INC - Number of cells empty after compression

Examples

COMPRB is called by ABOXST when new materiel is to be added and there is no new cell available.

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/

TSTEP - time step

Program Inputs

From Calling Program.

Arguments - all except INC, see above

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Arguments -

X - Array of Boxcar cells

XW - Current width of a cell (in time steps)

FOLD - Fraction of oldest cell currently occupied

INC - Number of cells empty after compression

Programs Called

Verbs. None

Other. None

Input/Output Files Used

None.

DBOXST
R/SS/CS
3/74

DBOXST (DSTN, CDS, WT)

General Description

Returns as DBOXST the average time the oldest amount WT currently in boxcar DSTN(CDS) has been in it, and removes WT and updates IOLD, FOLD, TOLD, and NCUR accordingly. Intended as a device for collecting delay time statistics in a rate/level model. The boxcar data are assumed stored in a PDS dataset DSTN(CDS) as follows.

Element	Name	Definition
1	IOLD,NMAX	Index in X of oldest box, and dimension of X
2	INEW,NCUR	Index in X of newest box, and no. boxes in use
3	XW	Width of a box, in time steps
4	TOLD	Oldest time in current content of IOLD
5	FOLD	Fraction of oldest box currently occupied
6	X(4)	
7	X(5)	
8	X(6)	Assumes IOLD=5, INEW=4, NMAX=NCUR=7
9	X(7)	
10	X(1)	X(K) is the amount whose current age is about
11	X(2)	TNOW-(XW*TSTEP*(NMAX-K))
12	X(3)	

Arguments.

DSTN - dataset type name of boxcar to remove materiel from

CDS - coordinates of boxcar dataset

WT - weight to be removed from boxcar

Examples

DBOXST is called when receipts come in to a node to determine the age of the receipts.

GASP Files Used

None.

(DBOXST-1)

Permanent Attributes Accessed

In/RLSYS/:

TSTEP - time step

In/VRBGSP/:

TNOW - current simulation time

In/DSNOW/:

IXDST - index of dataset type name and associated pointers

IXEL1 - index in DSPOOL of first element in dataset of current interest

PDS Datasets:

DSTN(CDS) - boxcar dataset, all elements as in general description.

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

The revised boxcar data are stored in PDS dataset DSNT(CDS)

To Calling Program.

DBOXST - average elapsed time since the removed resources were entered into the dataset

Programs Called

Verbs. None

Other. CLEAR, GETSET, IPAKIT, IUNPAK, PERMDS, SETSET

Input/Output Files Used

NPRNT - Print File

DLYLEV (DLY, NEL)

General Description

Returns to calling program the content of the delay whose dataset is in DLY.

Arguments.

DLY - delay dataset array

NEL - dimension of array DLY

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RL(1) - IXDPAR, 1st element of system dynamics delay of current interest

RLDPAR(1,J) - parameters of Jth system dynamics delay, code indicating form of delay

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Function Value -

DLYLEV - resource content of delay in DLY

Programs Called

Verbs. None

Other. None

Input/Output Files Used

None.

THIS PAGE LEFT INTENTIONALLY BLANK

DMAP
R/SS/CS
3/75

DMAP (RLDOLD, RLDNEW,A)

General Description

Maps fractions A of old RL delay with coordinate RLDOLD into RL delay with coordinate RLDNEW. If the old and new delays are of the same type and parameters, they can be mapped directly. Otherwise, DTIM and RLDREV are used to construct cells with the old content redistributed to correspond to the new delay form.

Arguments.

RLDOLD - coodinate of old RL delay

RLDNEW - coordinate of new RL delay

A - fraction of old RL delay mapped into new RL delay

GASP File Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RLDST - list of PDS dataset type names

RL - elements of system dynamics delay currently being updated

RLDPAR - parameters for system dynamics delay

In/TBLCOM/:

DTABLE - array for storing pairs of values, parameters for boxcar delays.

In/DSNOW/:

NE - number of elements in dataset of current interest

IXEL1 - index in DSPOOL of first element in dataset of current interest

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

PDS Datasets:

RLD (RLDNEW) - elements of system dynamics delay that have been updated

To DTIM.

Arguments -

First - index to delay parameters in RLDPAR

Second - array of times for RLD cells

To RLDREV.

Arguments -

First - array of old cell values

Second - array of times for old cells

Third - number of cells in old delay

Fourth - type of old delay

Fifth - array of new cell values

Sixth - array of times for new cells

Seventh - number of cells in new delay

To Calling Program.

None.

Programs Called

Verbs. None

Other. CLEAR, DLYLEV, DTIM, GETSET, PERMDS, RLDREV, SETSET

Input/Output Files Used

None.

DTIM (IDLY, T)

General Description

Constructs table of cumulative times, T, associated with ends of cells in delay whose index in RLDPAR is IDLY.

Arguments.

IDLY - index of RLDPAR of delay of current interest

T - table of cumulative times

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RLDPAR(*,J) - parameters for Jth system dynamics delay, J = 1, 2, ..., NRLDPR, as follows:

RLDPAR(1,J) - code indicating form of delay

RLDPAR(2,J) - mean delay time for 1st and 3rd order delay; pointer to DTABLE for boxcar delays.

In/TBLCOM/:

DTABLE - array for storing pairs of values, in this case the boxcar delay values in table form.

Program Inputs

From Calling Program.

Argument IDLY - see above

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Argument:

T - table of cumulative times associated with ends of cells in delay whose index in RLDPAR is IDLY.

Programs Called

Verbs. None

Other. None

Input/Output Files Used

None.

IRLC (RDSTC, NRDSTC)

General Description

Searches table of dataset coordinates for rate dataset whose coordinates are RDSTC. NRDSTC is the number of coordinates for the current dataset. The table of coordinates searched is array RLC in /LC/. Returns index in RLC of RDSTC, or zero if not present.

Arguments.

RDSTC - coordinates of rate dataset to be sought in table of dataset coordinates

NRDSTC - number of coordinates for current dataset

GASP Files Used

None.

Permanent Attributes Accessed

In /LC/:

RLC (*,I) - Ith unique set of PDS coordinates for datasets that represent or own system dynamics flow path

NRLC - maximum number of entries permitted in RLC (2nd dimension of RLC)

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes. None.

To Calling Program.

Function Value - IRLC - index in RLC of dataset coordinates RDSTC, or zero if not present

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

None.

THIS PAGE LEFT INTENTIONALLY BLANK

ISTAKR (ITOP, NOD, IPRED)

General Description

Returns index in IZHOLD of entry whose node number is NOD in stack whose top is at ITOP. Also returns index of predecessor in stack - as IPRED.

Arguments.

ITOP - top location of stack in IZHOLD

NOD - node number in stack in IZHOLD

IPRED - index of predecessor to NOD in stack in IZHOLD

GASP Files Used

None.

Permanent Attributes Accessed

In /PDSSYS/:

IPAK - packing factor

In Blank Common:

IZHOLD - pushdown stacks array

Program Inputs

From Calling Program.

Arguments -

ITOP - see above

NOD - see above

Program Outputs

To Permanent Attributes. None.

To Calling Program.

Argument IPRED - index of predecessor to NOD entry in IZHOLD

ISTAKR - index in IZHOLD of entry whose node number is NOD

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

None.

THIS PAGE LEFT INTENTIONALLY BLANK

IXLCIX (N)

General Description

Assigns next N locations available in LCIX. Returning the index of the first location as IXLCIX. If N are not available at the end of currently assigned space but enough can be reclaimed from freed space by garbage collection, garbage collection is implemented through LCIXGC.

Arguments.

N - number of locations to be assigned in LCIX.

GASP Files Used

None.

Permanent Attributes Accessed

In /LC/:

NLCIX - dimension of LCIX

NXTLCX - index of next available position in LCIX

NFRELC - number of positions in LCIX that have been freed for reuse

Program Inputs

From Calling Program.

Argument - see above

From LCIXGC.

In /LC/:

NXTLCX - index of next available position in LCIX

Program Outputs

To Permanent Attributes.

In /LC/:

NXTLCX - index of next available position in LCIX

To LCIXGC. None.

To Calling Program.

IXLCIX - index of first location in LCIX where N locations have been assigned

(IXLCIX-1)

Programs Called

Verbs. None.

Other. LCIXGC, ZERRPT.

Input/Output Files Used

NPRNT - Print file.

(IXLCIX-2)

LCIXGC

General Description

Garbage collection in LCIX in /LC/. Consolidates current entries at front of array and sets NXTLCX to first entry following them. Also loops through LDSC datasets to update pointer into LCIX from associated lists of LKDSTS. LCIX is assumed to contain, in addition to the current (positive) entries to be preserved, blocks of released space, each of which contains the negative count of its length in the first position and zeros in the remainder.

Arguments.

None.

Examples

LCIXGC is called by IXLCIX when sufficient contiguous space is not available in LCIX, but freed space is available.

GASP Files Used

None.

Permanent Attributes Accessed

In /LC/

- LCIX - array of pointers into RLC
- NLCIX - dimension of LCIX
- NXTLCX - index of next available position in LCIX
- NFRELC - number of positions in LCIX that have been freed for reuse

In /RLSYS/

- RLDST - list of PDS dataset type names containing elements potentially subject to system dynamics statistics collection
- NRLN - number of system dynamics nodes in current model

In /TBLCOM/

- DTABLE - array for storing pairs of values

Program Inputs

From Calling Program. None.

From NEWLCX.

Function Value - newly displaced pointer to LCIX

Program Outputs

To Permanent Attributes.

In /LC/

LCIX - array of pointers into RLC

NXTLCX - index of next available position in LCIX

NFRELC - number of locations in LCIX freed for reuse

To Calling Program. None.

To NEWCLX.

Argument

IDTABL (2, IDT) - current pointer to LCIX (before LCIX garbage collection)

NDISPL - displacements caused by LCIXGC

NJ - index to NDISPL array to search up to.

Programs Called

Verbs. None.

Other. CLEAR, IUNPZ, NEWLCX, PERMDS, VALEL.

Input/Output Files Used

NPRNT - Print File.

(LCIXGC-2)

LRLOOP (RDSTYP, RATPS)

General Description

Loops through flow-path-owning PDS datasets of type RDSTYP. For each RDSTYP dataset that owns paths containing RLNOD, the current system dynamics node number in /RLSYS/, the elements of the dataset are returned as argument RATPS, and the index in RLC of the coordinates of the dataset is returned as the function value, LRLOOP. The loop is implemented by stepping through the block of RLC pointers in LCIX that are associated with dataset type RDSTYP in the list of dataset types pointed to from LDSC(RLNOD).1. When the loop is finished, LRLOOP = 0 is returned. LRLOOPs may not be nested.

Arguments.

RDSTYP - PDS dataset type name, belonging to LDSC (RLNOD), whose RLC pointers are to be looped through

RATPS - array in which elements of dataset are returned to calling program

Examples

An important user of LRLOOP is vRLOOP, which facilitates the consideration of all flowpaths of a particular type that feed a node.

GASP Files Used

None.

Permanent Attributes Accessed

In PDS Datasets:

LDSC(RLNOD).1 - pointer to list of dataset types owning flow paths of which RLNOD Is a member

RDSTYP(RLC(ldsc(rlnod))) - all datasets of type RDSTYP owning at least one flow path of which RLNOD is a member

In/LC/:

LCIX - pointers to RLC

RLC - coordinate sets

In/TBLCOM/:

DTABLE - lists of dataset types

Program Inputs

From Calling Program.

Arguments, described above. Also, RLNOD and RLMODE in /RLSYS/.

From PERMDS (VALEL, LDSC, RLNOD, I, DSTIX).

Argument DSTIX, pointer to list of dataset types owning paths containing RLNOD. Also, in /DSNOW/, IXDST ≦ 0 is LDSCs not defined, or IXELI ≦ 0 if LDSC(RLNOD) not found.

From ISTAK (IDSI, I, RDSTYP).

Function value, LRLOOP, location in IZHOLD of RDSTYP for current LDSC, or 0 to signal finish.

From IXDTI (IDSTIX, RDSTYP).

Function value, LRLOOP, location in DTABLE of RDSTYP for current LDSC.

From PERMDS (GETSET, RDSTYP, RLC(I, LRLOOP), 0, RATPS).

Argument RATPS, elements of RDSTYP (RLC(I, LRLOOP)). Also, in /DSNOW/, IXELI = -1 if dataset does not exist.

From SAVLK (during setup mode only).

In /LC/, LRLCHG = 1 if RLC pointers being scanned have been switched from LCIX residency to push down stack.

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

As function value, LRLOOP, location in RLC of current coordinates; and, as argument RATPS, elements of current RDSTYP dataset.

(LRLOOP-2)

To PERMDS (VALEL, LDSC, RLNOD, I, DSTIX).

All arguments except last.

To ISTAK (IDSI, I, RDSTYP).

Arguments: IDSI, index in IZHOLD of top of current dataset types pushdown stack; I, word of stack entries to be searched; RDSTYP, value sought.

To IXDTI (IDSTIX, RDSTYP).

Arguments: IDSTIX, LDSC pointer to DTABLE list of dataset type names; RDSTYP, dataset type name being sought.

To PERMDS (GETSET, RDSTYP, RLC(I, LRLOOP), 0, RATPS).

Arguments identifying RDSTYP (RLC(*,LRLOOP)) as dataset to be retrieved.

To SAVLK.

In /LC/: LRLPTR, index in IZHOLD or DTABLE of current RLC pointer, or -1 when LRLOOP is idle; LRLCHG = 1 if current RLC pointer loop is in IZHOLD, LRLCHG = 0 if in DTABLE.

Programs Called

Verbs. None

Other. GETSET, ISTAK, IUNPKZ, IXDTI, PERMDS, VALEL

Input/Output Files Used

None.

LRNIX (LRNAM)

General Description

Returns index of name LRNAM in array LRDSTN, whose entries are names of linked-rate dataset types.

Arguments.

LRNAM - name in array LRDSTN

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/

RLDST - list of PDS dataset type names

NRDTMX - dimension of array RLDST

NRLDST - number of entries in RLDST currently used

Program Inputs

From Calling Program.

Argument - see above

Program Outputs

To Permanent Attributes.

In /RLSYS/:

RLDST - list of PDS dataset type names - new name

NRLDST - number of entries in RLDST currently used - increased by 1 if LRNAM added to RLDST

To Calling Program.

Function Value - index of name LRNAM in array LRDSTN

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

NPRNT - Print File.

NEWLCX (IPTRW, NDISPL, NJ)

General Description

Returns as NEWLCX the value of IPTRW with its right-hand portion revised to reflect the changes in LCIX positions caused by garbage collection in LCIX. A table of displacements is assumed to be in NDISPL. An entry in the table is of the form (D, I), where I is the first position in LCIX whose content was displaced by an amount D. The first entry is always (0, 1).

Arguments.

IPTRW - old pointer to LCIX array, before LCIXGC

NDISPL - table of displacements caused by garbage collection

NJ - entry in NDISPL to search up to

GASP Files Used

None.

Permanent Attributes Accessed

None.

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes. None.

To Calling Program.

NEWLCX - value of IPTRW with its right-hand portion revised to reflect the changes in the LCIX positions caused by garbage collection in LCIX

Programs Called

Verbs. None.

Other. IPAKZ, IUNPKZ.

Input/Output Files Used

None.

(NEWLCX-1)

THIS PAGE LEFT INTENTIONALLY BLANK

NEWRLC (RLDST, NRLDST)

General Description

Stores coordinates RLDST in next available location in RLC in /LC/, returning as NEWRLC the index of this location. A fatal error is incurred if there is no more room in the table.

Arguments.

RLDST - coordinates of rate/level dataset type to be entered in RLC array

NRLDST - number of coordinates for rate/level dataset type

GASP Files Used

None.

Permanent Attributes Accessed

In /LC/:

NXTRLC - index of next available position in RLC

NRLC - maximum number of entries permitted in RLC (second dimension of RLC)

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

In /LC/:

RLC (*,I) - I-th unique set of PDS coordinates for datasets that represent or own system dynamics flow paths

To Calling Program

NEWRLC - index of location where coordinates RLDST are stored in RLC in /LC/

Programs Called

Verbs. None.

Other. ZERRPT.

Input/Output Files Used. NPRNT - Print File.

THIS PAGE LEFT INTENTIONALLY BLANK

PRRLST

General Description

Prints a rate/level statistic. IRLST in /RLSYS/ is the index in KRLST of the pointer to the statistic to be printed.

Arguments.

None.

GASP Files Used

None.

Permanent Attributes Accessed

In/DSTYPS/

NDST - number of entries in each of arrays DSTYP, DSTDES and COORD1

DSTYP - array of dataset type names

DSTDES - array of pointers to table descriptors and dataset type descriptors

In/RLSYS/

PRLSSW - page header control switch

IXDSPL - index in DSPOOL of value to be observed for current statistic

MRLST - index of entry in KRLST1 corresponding to KRLST (IRLST)

In/STNAM/

RLSNAM - array of Rate/Level statistic names

Program Inputs

From Calling Program.

None.

Program Outputs

To Permanent Attributes.

None

To File NPRNT.

Identifiers and current value of a system dynamics variable statistic

To Calling Program.

None.

Programs Called

Verbs. None

Other. HASHI, IUNPAK, IUNPKZ, LIN, MSHIFT

Input/Output Files Used

NPRNT - Print File

RELCIX (N, I1)

General Description

Releases N words from the array of pointers LCIX beginning at location I1.

Arguments.

N - number of words to be released from LCIX

I1 - first location to begin release of N words from LCIX

GASP Files Used

None.

Permanent Attributes Accessed

In /LC/:

LCIX - array of pointers in RLC

NXTLCX - index of next available position in LCIX

NFRELC - number of positions in LCIX that have been freed for reuse

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

In /LC/:

LCIX - array of pointers into RLC

NFRELC - number of positions in LCIX that have been freed for reuse

To Calling Program. None.

Programs Called

Verbs. None.

Other. CLEAR.

Input/Output Files Used

NPRNT - Print File.

(RELCIX-1)

THIS PAGE LEFT INTENTIONALLY BLANK

REVRLS

General Description

Updates DSPOOL pointers in KRLST1 to reflect the effect of DSPOOL garbage collection.

Arguments.

None.

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/:

KRLST(I) - partial identifier of K-th system dynamics statistical variable

NXRLST - last index in KRLST, etc., assigned

KRLST1(K)- remainder (after KRLST(I)) of identification of K-th system dynamics statistical variable

Program Inputs

From Calling Program. None.

From IXNEW.

Function value - IXNEW, the index in the revised DSPOOL corresponding to IDSPO before DSPOOL was revised through garbage collection

Program Outputs

To Permanent Attributes.

In /RLSYS/:

KRLST1(K) - remainder (after KRLST(I)) of identification of K-th system dynamics statistical variable - contains (STAT1 index *100000) + index in DSPOOL of value to be observed

To IXNEW

Argument

(REVRLS-1)

IDSPO - index in DSPOOL of a dataset before garbage collection was performed

To Calling Program. None.

Programs Called

Verbs. None.

Other. IXNEW.

Input/Output Files Used

None.

(REVRLS-2)

RLDLY
R/SS/CS
1/74

RLDLY (DLRIX, RATEIN)

General Description

General delay for Rate/Level logic. For the rate whose current value is RATEIN and whose status for the delay of current interest is stored in PDS RLD (DLRIX), RLDLY returns the updated output rate, obtained by branching to the function representing the specific type of rate associated with DLRIX. This association occurs through RLD (DLRIX).1, which contains codes 1, 2, . . ., 6 for delay types

1 - 1ST ORDER EXP, INFO
2 - 1ST ORDER EXP, MATERIAL
3 - 3RD ORDER EXP, INFO
4 - 3RD ORDER EXP, MATERIAL
5 - ARBITRARY, INFO
6 - ARBITRARY, MATERIAL

These delays follow Forrester's form in Industrial Dynamics

Arguments.

DLRIX - index representing a specific delay for a specific rate

RATEIN - current rate in

Examples

The function RLDLY is used whenever a rate is to be delayed in a module.

DLYRAT = RLDLY (INDEX, RATE)

where RATE is the new rate, INDEX is the index to the rate/level delay to be used, and DLYRAT is the delayed rate. DLYRAT can then be used to update the demand or receipt stack.

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

RLD - RLDST(12) 12-th entry in list of PDS dataset type names

RL - array of elements of system dynamics delays

DLRIXK - last index assigned as the coordinate of an RL delay

(RLDLY-1)

RLDPAR (*, J) parameters for Jth system dynamics delay (J = 1, 2 . ., NRLDPR), as follows:

RLDPAR (1, J) - code indications form of delay

RLDPAR (2, J) - mean delay time for 1st and 3rd order delay; pointer to DTABLE for boxcar delays.

Program Inputs

From Calling Program.

Arguments - see above

From RLID1 function value, RLID1, updated output rate of 1st order exponential information delay

From RLMD1 function value, RLMD1, updated output rate of 1st order exponential material delay

From RLID3 function value, RLID3, updates output rate of 3rd order exponential information delay

From RLMD3 function value, RLMD3, updated output rate of 3rd order exponential material delay

From RLIDB function value, RLIDB, updated output value of boxcar information delay

From RLMDB function value, RLMDB, updated output value of boxcar material delay

Program Outputs

To RLID1, RLMD1, RLID3, RLMD3, RLIDB, RLMDB

Arguments

DLRIX - index representing a specific delay of a specific rate

RATEIN - current rate in

To Permanent Attributes. None.

To Calling Program.

Function Value -

RLDLY - output rate of delay specified in RLD (DLRIX) after updating it for current time step

Programs Called

Verbs. None.

Other. ARLDIX, GETSET, PERMDS, RLIDB, RLID1, RLID3, RLMDB, RLMD1, RLMD3.

Input/Output Files Used

6 - system print file

THIS PAGE LEFT INTENTIONALLY BLANK

RLDREV
R/SS/CS
1/75

RLDREV (XO, TO, NOLD, INCO, XN, TN, NNEW)

General Description

Revises levels in RL delays to reflect new parameters. If the old delay time is longer than the new time, the materiel is expanded across the cells. If the old delay time is shorter than the new time, the materiel is compressed in the cells.

Arguments.

XO - Array of old levels

TO - Cumulative times associated with old cells

NOLD - Number of cells in old delay

INCO - 2 for material boxcars, 1 otherwise

XN - Array of new levels

TN - Array of new cumulative times

NNEW - Number of cells in new delay

GASP Files Used. None.

Permanent Attributes Accessed. None.

Program Inputs

From Calling Program.

Arguments - see above.

Program Outputs

To Permanent Attributes. None.

To Calling Program.

Arguments:

XN - Array of new levels

TN - Array of new cumulative times

NNEW - Number of cells in new delay

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used. None.

(RLDREV-1)

THIS PAGE LEFT INTENTIONALLY BLANK

RLDRPT

General Description

Prints report of Rate/Level delays. An example of the report is shown in the accompanying figure.

Arguments. None.

Examples. RLDRPT is a routine, but it may be referenced in a model description through the verb ORLD.

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

RLDPAR (*, J) parameters for Jth system dynamics delay, J = 1, 2, . . ., NRLDPR, as follows:

RLDPAR (1, J) - code indicating form of delay

RLDPAR (2, J) - mean delay time for 1st and 3rd order delay; pointer to DTABLE for boxcar delays

RLDNAM (*, J) alphanumeric description of Jth delay (J = 1, 2, . . ., NRLDPR)

NRLDPR - maximum number of system dynamics type delays permitted in current model

RLDTYP - alpha identifier of rate/level delay type

Program Inputs

From Calling Program. None.

Program Outputs

To Permanent Attributes. None.

To Calling Program. None.

Programs Called

Verbs. None.

Other. LIN.

Input/Ouput Files Used

NPRNT - Print File

(RLDRPT-1)

```
** MAWLOGS MODEL DSS2          **                  ANALYST.,BURGER,R.T
                                                   DATE.      1/ 14/ 1975
          SUMMARY OUTPUT                           RUN NAME.VII CORPS
                                                   TNOW        90.000

                     RATE/LEVEL DELAYS

     INDEX      TYPE       PARAMETER

****   1       RLMD3         18.000    ACCT-GSUV CST, HI PRI                        ****
****   2       RLMD3         40.000    ACCT-GSUV CST, LO PRI                        ****
****   3       RLMD3         18.000     ACCT-GSUW CST, HI PRI                       ****
****   4       RLMD3         40.000    ACCT-GSUW CST, LO PRI                        ****
****   5       RLMD3         18.000    ACCT-GSUF CST, HI PRI                        ****
****   6       RLMD3         40.000    ACCT-GSUF CST, LO PRI                        ****
****   7       RLMD1          6.500    CONUS DEPOT TO CCP IN-TRANSIT TIME           ****
****   8       RLMD3      90001.000    TACOM (MCLAS 1)  CONUS FILL, HI PRI          ****
                                           CUMD        D         P(D)    CUMP(D)
                                          13.000    13.000      .0001     .0001
                                          18.000     5.000      .5823     .5824
                                          43.000    25.000      .1150     .6974
                                          73.000    30.000      .0805     .7779
                                         103.000    30.000      .0493     .8272
                                         133.000    30.000      .0405     .8677
                                         163.000    30.000      .0216     .8893
                                         193.000    30.000      .0186     .9079
                                         253.000    60.000      .0921    1.0000
                                 MEAN D   53.365
****   9       RLMD3      90010.000    TACOM (MCLAS 1)  CONUS FILL, LO PRI          ****
                                           CUMD        D         P(D)    CUMP(D)
                                          13.000    13.000      .0001     .0001
                                          18.000     5.000      .5827     .5828
                                          43.000    25.000      .0849     .6677
                                          73.000    30.000      .0665     .7342
                                         103.000    30.000      .0536     .7878
                                         133.000    30.000      .0379     .8257
                                         163.000    30.000      .0360     .8617
                                         193.000    30.000      .0339     .8956
                                         253.000    60.000      .1044    1.0000
                                 MEAN D   59.311
****  10       RLMD3      90019.000      DSA   (MCLAS 2)  CONUS FILL, HI PRI        ****
                                           CUMD        D         P(D)    CUMP(D)
                                          13.000    13.000      .0001     .0001
                                          18.000     5.000      .8647     .8648
                                          43.000    25.000      .0446     .9094
                                          73.000    30.000      .0224     .9318
                                         103.000    30.000      .0122     .9440
                                         133.000    30.000      .0103     .9543
                                         163.000    30.000      .0071     .9614
                                         193.000    30.000      .0018     .9632
                                         253.000    60.000      .0368    1.0000
                                 MEAN D   27.930
```

FIGURE - SAMPLE REPORT PAGE FROM RLDRPT

RLIDB
R/SS/CS
1/74

RLIDB (DLRIX, RATEIN)

General Description

Function which updates the output of a delayed rate of arbitrary form. The delayed rate updated is that represented by DLRIX. Its current input value is RATEIN. The delay distribution simulated is that represented by the probability table in DTABLE pointed to from RLD (DLRIX).2. The probability table consists of a sequence of couples (P, D), where P is a probability and D is an increment of the total delay. The entries in the table are assumed to represent increasing cumulative delay as one moves downward in the table. (This is Forrester's boxcar delay).

Arguments.

DLRIX - index representing a specific delay of a specific rate

RATEIN - current rate in

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

TSTEP - time step

NSTEP - number of time steps executed to date

NSTW - number of time steps to be used for accelerated warmup of flow rates subject to delay

RLDST - array of PDS dataset type names

RL - elements of system dynamics delay currently being defined or updated

RLDPAR - parameters for system dynamics delay

Program Inputs

From Calling Program.

Arguments - see above

In /RLSYS/:

RL - array of elements of the dataset of dataset type RLD and coordinate DLRIX

(RLIDB-1)

<u>Program Outputs</u>

<u>To Permanent Attributes</u>. None.

<u>To Calling Program</u>.

Function Value -

RLIDB - updated output value of boxcar information delay

<u>Programs Called</u>

<u>Verbs</u>. None.

<u>Other</u>. GETSET, PERMDS, SETSET.

<u>Input/Output Files Used</u>. None.

RLID1 (DLRIX, RATEIN)

General Description

Function which returns updated output rate for the rate represented by DLRIX, whose current input rate is RATEIN. This is Forrester's 1st order exponential delay for information (The Smooth Macro In DYNAMO).

Arguments.

DLRIX - index representing a specific delay of a specific rate

RATEIN - current input rate

GASP Files Used. None.

Permanent Attributes Accessed

In /RLSYS/:

TSTEP - time step, in number of simulation time units

RLDST - array list of PDS dataset type names

RL - array of elements of system dynamics delay currently being defined or updated

RLDPAR - parameters for system dynamics delay

Program Inputs

From Calling Program.

Arguments - see above

In /RLSYS/:

RL - array of elements of dataset type name RLD and coordinate DLRIX

Program Outputs

To Permanent Attributes. None.

To Calling Program.

Function Value -

RLID1 - updated output rate of first order exponential information delay

Programs Called

Verbs. None.

THIS PAGE LEFT INTENTIONALLY BLANK

RLID3 (DLRIX, RATEIN)

General Description

Function which returns updated value of output of the delayed rate represented by DLRIX. The delay function represented is Forrester's 3rd order information delay (Macro DLINF3 in Dynamo).

Arguments.

DLRIX - index representing a specific delay of a specific rate

RATEIN - current input rate

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

TSTEP - time step

NSTEP - number of time steps executed to date

NSTW - number of time steps to be used for accelerated warmup of flow rates subject to delay

RLDST - array of PDS dataset type names

RL - elements of system dynamics delay currently being defined or updated

RLDPAR - parameters for system dynamics delay

Program Inputs

From Calling Program.

Arguments - see above

In/RLSYS/:

RL - array of elements of the dataset of type RLD and coordinate DLRIX

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Function Value -

RLID3 - updated value of output rate of 3rd order exponential information delay

Programs Called

Verbs. None

Other. GETSET, PERMDS, SETSET

Input/Output Files Used

None.

RLINT (DSETNC, VELC)

General Description

Changes certain elements of RLPDS datasets to integer values. This routine is called by RLPDS after an input card is read in.

Arguments.

DSETNC - dataset name

VELC - array of elements to change to integer values

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/

RLDST - list of PDS dataset type names containing elements potentially subject to system dynamics statistics collection

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Argument VELC - array of elements as input, but in integer form for specified datasets: MGTCL, OBOX, EXRAT, FCRAT, PRIORY for VELC(1); EXRAT, FCRAT, OBOX for VELC(2).

Programs Called

Verbs. None

Other. None

Input/Output Files Used

None.

THIS PAGE LEFT INTENTIONALLY BLANK

RLLNK (NRLNOD, IXRATE, RATE, DLY)

General Description

Principal builder of system dynamics flow path network - accepts link definitions from vRLDLK and vRLRLK and changes from vADDLK, vADRLK, vCHDLK, vCHRLK, vDEDLK, and vDERLK. An execution of RLLNK concerns a single flow link, namely the one whose destination is node NRLNOD in the path pointed to from PDS element LKDST(LKCDS).IXRATE.

The flow path network and changes to it are accumulated in pushdown stacks in IZHOLD. Access paths for member nodes are built by creating LDSC datasets, and for each LDSC (node) dataset a pushdown stack of dataset type names owning one or more paths in which the node has membership. Specific datasets in whose paths the node has membership are identified by pushdown stacks, one for each dataset type name, whose entries are pointers to entries in RLC, the list of owner coordinate sets. These data structures are described in detail in Chapter III. RLLNK has the overall responsibility for creating pushdown stacks, maintaining pointers to them, adding new coordinate sets to RLC, creating LDSC datasets for new nodes, etc. These activities are outlined in their execution sequence in the accompanying figure. Note that when changes to an existing flow path are to be made, RLLNK converts the path from reference form to setup form (step 7 in the logic outline).

RLLNK requires that variables LKDST and LKCDS in /RLSYS/ have been set in the host modules, and that PDS element LKDST(LKCDS).IXRATE exist. If the dataset element does not exist, no path for it can be defined.

The role of RLLNK is limited to the accumulation of network data in setup form. The model writer is responsible for insuring that vRLFLK or a suitable substitute is subsequently executed to convert the setup form to reference form for use in the simulation.

Arguments.

NRLNOD - name of system dynamics node to be added to the flow network or whose link identifies or is affected by a change

(RLLNK-1)

IXRATE - index in PDS dataset LKDST (LKCDS) of element that points to flow path of current interest

RATE - ≥0, flow rate to be assigned to new link

<0, change code: -1, delete the link into NRLNOD

-2, insert a link after an existing one

-3, substitute a new link for an existing one

-4, insert a link before an existing one

DLY - index of system dynamics delay form and parameters to be used on current link.

Examples

The flow of data to RLLNK will be illustrated by a reference to vRLDLK to define a demand link to node A in the resource X flow path owned by node H via the 4th element of dataset type PATHS. Module, vHOST is the simulation module in which PS1 is used to define the portion of the flow path network directly related to H. The following fragment of model description could be used:

```
H.RLNOD (*H, 19.),
    HOST (1 = RLDLK (P = *A, 11.)
```

Key lines in HOST would include:

```
    COMMON/RLSYS/RLNOD, LKDST, LKCDS(5), IXDRAT, DRATE, ...
C SET /RLSYS/ FOR FLOW PATH DEF IN PS1
    LKDST = "PATHS"
    LKCDS(1) = RLNOD
    LKCDS(2) = X
    IXDRAT = 4
    DRATE = 0.
```

RLLNK is referenced in vRLDLK as follows:

```
    COMMON/RLSYS/..., IXDRAT, DRATE, ...
    CALL RLLNK (NRLNOD, IXDRAT, DRATE, DLY)
```

where NRLNOD and DLY are the arguments of vRLDLK, namely "A" and 11., respectively.

(RLLNK-2)

Statistics Collected

None.

GASP Files Used

None.

Permanent Attributes Accessed

PDS Elements.

LKDST(LKCDS).IXRATE - flow path owner pointer

LDSC(RLNIX(NRLNOD)).1 - pointer to list of dataset types owning paths of which NRLNOD is a member

COMMON /RLSYS/.

RLC - array of coordinate sets of path-owning datasets

NRLC - number of entires currently in RLC

RATDST - list of dataset types that own flow paths in this model

NRATDT - number of names currently in RATDST

Program Inputs

From Calling Program.

Arguments as defined above.

In Common /RLSYS/ -

LKDST - name of path-owning dataset type

LKCDS - coordinates of path-owning dataset

From PERMDS (VALEL, LKDST, LKCDS, IXRATE, RDSP)

Argument RDSP - pointer to flow path of interest if it exists

Common /DSNOW/ -

IXDST - >0 if LKDST exists, $\leq$0 if not

IXELI - $\geq$0 if LKDST(LKCDS).IXRATE exists, <0 if not

From RLNIX (NRLNOD).

Function value, RLNODE, the system dynamics node number of node NRLNOD

From PERMDS (VALEL, LDSC, RLNODE, 1, RDSTP).

Argument RDSTP - pointer to list of dataset type names of datasets owning paths of which NRLNOD is a member, if dataset exists

(RLLNK-3)

Common /DSNOW/ -

IXELI - <0 if LDSC (RLNODE) does not exist, $\geq$0 otherwise

From 1st ZPLAEN. IZREF in /ZSTACK/, index in IZHOLD of top of stack containing LKDSTs for current LDSC.

From PERMDS (ADDSET, LDSC, RLNOD, 1, RDSTP). None.

From SAVLK (IRDSTP). Argument IRDSTP - negative pointer into IZHOLD of stack of LKDSTs for current LDSC. Also associated RLC pointers in pushdown stacks.

From PERMDS (SETEL, LDSC, RLNODE, 1, IRDSTP). None.

From NDIDST (LKDST). Function value, NLKCDS, the number of coordinates for LKDST datasets.

From IRLC (LKCDS, NLKCDS). Function value, JCDS. If >0, index of coordinate set LKCDS in RLC. If $\leq$0, indicator that LKCDS are not in RLC.

From NEWRLC (LKCDS, NLKCDS). Function value, JCDS, index of position of LKCDS just added to RLC.

From ISTAK (IX, 1, LKDST). Function value, JDST. If >0, index in IZHOLD of LKDST. If $\leq$0, indicator that LKDST is not in pushdown stack of dataset type names for current LDSC.

From 2nd ZPLAEN. IZREF in /ZSTACK/, index in IZHOLD of top of new pushdown stack of RLC pointers associated with current LKDST.

From 3rd ZPLAEN. IZREF in /ZSTACK/, saved as ILKP, index in IZHOLD of top of new pushdown stack for LKDSTs for current LDSC.

From PERMDS (SETEL, LDSC, RLNODE, 1, RDSTP). None.

From ISTAKI (IZREF, 2, JCDS). Function value, ILKP. If >0, index in IZHOLD of entry in stack of RLC pointers that points to LKCDS. If $\leq$0, indicator that entry is not in current stack.

From 4th ZPLAEN. IZREF in /ZSTACK/, saved as ILKP, index in IZHOLD of entry in stack RLC pointers that has just been added to point to LKCDS.

From TRSTAK (IRDSP). Function value, saved as IRDSP, index in IZHOLD of entry that points to pushdown stack containing current flow path.

(RLLNK-4)

From 5th ZPLAEN. IZREF IN /ZSTAK/, saved as -IRDSP, index in IZHOLD of entry that points to pushdown stack containing current flow path.

From PERMDS (SETEL, LKDST, LKCDS, IXRATE, RDSP). None.

From 6th ZPLAEN. IZREF in /ZSTACK/, index in IZHOLD of entry just added to new flow path.

From ZREMEN (at statement 230). IZREF, saved as IZRFN, index in IZHOLD of current top of new flow path. Also, in IZKC and IZK in /ZSTACK/ content of previous top of stack for path.

From 7th ZPLAEN. IZREF, index in IZHOLD of new location for stack entry just removed by ZPLAEN, now stored as first entry in a stack that will have two entries to describe the change.

From 8th ZPLAEN. IZREF, index in IZHOLD of top entry of two-entry stack for a change.

From CPLAEN. IZREF, index in IZHOLD of new bottom entry in stack representing changes to current path. This entry points to stack created in 7th and 8th ZPLAENs.

From ZERRPT. None.

Program Outputs

To Permanent Attributes.

PDS Elements.

LKDST(LKCDS).IXRATE - negative pointer to IZHOLD entry pointing to pushdown stack containing flow path, if path had not previously been in IZHOLD.

LDSC(RLNIX(NRLNOD)).1 - negative pointer to pushdown stack of dataset types owning paths of which NRLNOD is a member.

Common /RLSYS/.

RLC - LKCDS added if not previously present

NRLC - incremented by one if LKCDS was added

RATDST - LKDST added if it was not previously there

NRATDT - incremented by one if LKDST added.

(RLLNK-5)

Common /LC/ (via SAVLK).

LCIX - segments affected by changes have been released

NFRELC - count of LCIX entries that have been released is incremented if any were released

Common /TBLCOM/ (via TRSTAK).

DTABLE - segments freed if occupying paths were revised

NFREDT - number of freed entries - may be increased.

To Calling Program. None.

To PERMDS (VALEL, LKDST, LKCDS, IXRATE, RDSP). Arguments other than RDSP.

To RLNIX (NRLNOD). Argument NRLNOD, name of node.

To PERMDS (VALEL, LDSC, RLNODE, 1, RDSTP). Arguments other than RDSTP.

To 1st ZPLAEN. In /ZSTACK/ - IZKC = LKDST; IZK = 0.

To PERMDS (ADDSET, LDSC, RLNODE, 1, RDSTP). All arguments. RDSTP is negative pointer to stack of LKDSTs just created.

To SAVLK (IRDSTP). Argument IRDSTP - pointer to DTABLE - resident list of LKDSTs for current LDSC.

To PERMDS (SETEL, LDSC, RLNODE, 1, IRDSTP). All arguments. IRDSTP is revised pointer to LKDSTs list just returned by SAVLK.

To NDIDST (LKDST). Argument LKDST.

To IRLC (LKCDS, NLKCDS). Argument LKCDS.

To NEWRLC (LKCDS, NLKCDS). Arguments LKCDS, and NLKCDS, number of coordinates in LKCDS.

To ISTAK (IX, 1, LKDST). Arguments: IX, top of pushdown stack containing dataset types for current LDSC; 1, word of stack entries to be scanned; LKDST, value sought.

To 2nd ZPLAEN. In /ZSTACK/ - new RLC pointer entry

IZREF - 0, to start new stack

IZKC - 10_8 + NLKCDS, which is no. of coordinates in current LKDST, with a 1 packed (octal) to the left as the count of times node NRLNOD is referenced in paths owned by LKDST (LKCDS).

IZK - JCDS, location of LKCDS in RLC.

(RLLNK-6)

To 3rd ZPLAEN. In /ZSTACK/ - new LKDST entry

IZREF - IX, current top of LKDSTs stack for current LDSC

IZKC - LKDST

IZK - IPAK2 (1, IZREF), pointer to top of RLC pointers stack for current LKDST.

To PERMDS (SETEL, LDSC, RLNODE, 1, RDSTP). All arguments. RDSTP is negative pointer to current top of LKDSTs stack for current LDSC.

To ISTAKI (IZREF, 2, JCDS). Arguments:

IZREF - top of pushdown stack of RLC pointers for current LKDST

2 - word of stack entries to be scanned

JCDS - value to be found, namely index in RLC of current LKCDS.

To 4th ZPLAEN. In /ZSTACK/ - RLC pointer entry to current LKCDS.

IZREF - current top of stack of RLC pointers for current LKDST

IZKC - NLKCDS, number of coordinates in current LKCDS

IZK - JCDS, index in RLC of current LKCDS.

To TRSTAK (IRDSP). Argument IRDSP, pointer to DTABLE residence of current flow path.

To 5th ZPLAEN. In /ZSTACK/ - IZHOLD entry to store pointers to old and new flow path stacks.

To PERMDS (SETEL, LKDST, LKCDS, IXRATE, RDSP). All arguments. RDSP is stored as negative pointer to pointer entry created in 5th ZPLAEN.

To 6th ZPLAEN. In /ZSTACK/ - new flow path entry.

IZREF - top of current new flow path stack

IZKC - IPAK2 (IFIX(RLNODE), IFIX (DLY)), packed flow destination node number and link delay for new link

ZK - RATE, initial flow rate to be assigned link.

To ZREMEN (at statement 230). IZREF in /ZSTACK/, top of current new flow path stack.

To 7th ZPLAEN. In /ZSTACK/ - previous top of new flow path stack just obtained via previous ZREMEN, to become 1st entry in a two-member stack for current change.

(RLLNK-7)

To 8th ZPLAEN. In /ZSTACK/ - 2nd entry in change stack just initiated.

IZKC - packed node and delay of new or changed link

IZK - rate for new or changed link.

To CPLAEN. In /ZSTACK/ - change pointer entry for bottom of current new flow path stack

IZREF - current top of new flow path stack

IZKC - CHCOD

IZK - top of pushdown stack created by 7th and 8th ZPLAENs

To ZERRPT. None.

Programs Called

Verbs. None.

Other. ADDSET, CPLAEN, IPAKZ, IRLC, ISTAK, IUNPKZ, NDIDST, NEWRLC, PERMDS, RLNIX, SAVLK, SETEL, TRSTAK, VALEL, ZERRPT, ZSTAK.

Input/Output Files Used

NPRNT - print file, for error message.

1.	Check for existence of flow path pointer location LKDST(LKCDS).IXRATE. If not in model, print message and return.
2.	Insure that RATDST contains current LKDST.
3.	Set CHCOD = RATE if this is a change action, which is signalled by RATE < 0.
4.	Retrieve LDSC(RLNIX(NRLNOD)).1. Create the LDSC dataset if there is none, and initiate pushdown stack of associated LKDSTs.
5.	If LDSC.1 > 0, call SAVLK to remove associated LKDSTs from DTABLE to a pushdown stack, and the associated pointers to RLC from LCIX to pushdown stacks.
6.	Use IRLC to insure that LKCDS is in RLC. Use NEWRLC to add it if it isn't. Update count, in LKDST entry, of number of times node NRLNOD appears in paths owned by LKDST(LKCDS).
7.	If LKDST(LKCDS).IXRATE, the pointer to the flow path of current interest, is positive, use TRSTAK to transfer the path representation from DTABLE to a pushdown stack. Create a stack if this is the first entry in a new path.
8.	If CHCOD ≥ 0, not a change; add entry to top of pushdown stack representing new path. Then return.
9.	If CHCOD < 0, a change to an existing path is being described. So, instead of adding a link to a new path, store a change pointer at the <u>bottom</u> of the pushdown stack of change pointers and create as the object of the pointer a two-member pushdown stack that describes the change. Then return.

Outline of RLLNK Logic

(RLLNK-9)

THIS PAGE LEFT INTENTIONALLY BLANK

RLMDB
R/SS/CS
1/74

RLMDB (DLRIX, RATEIN)

General Description

Function which returns the current value of the output of the delayed rate represented by DLRIX, and updates the intermediate rates and levels stored in RLD (DLRIX). The delay simulated is one of the arbitrary distributions represented by the probability table stored in DTABLE at the locations pointed to from RLD (DLRIX).2. The probability table consists of couples (P,D), where P is a probability and D is an increment of the delay. The representation is that fraction P of the flow in the delayed rate which is delayed by the amount D(I)+D(I-1)+...+D(I), where the index of D gives its relative location in the probability table and the P in question is P(I). Neither the P nor the D tabulated are cumulative. (This is Forrester's material boxcar delay.)

Associated with each (P,D) entry in the probability table describing the delay distribution are an intermediate rate and level in RLD.

Arguments.

DLRIX - index representing a specific delay of a specific rate

RATEIN - current rate in

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

TSTEP - time step

NSTEP - number of time steps executed to date

NSTW - number of time steps to be used for accelerated warmup of flow rates subject to delay

RLDST - array of PDS dataset type names

RL - elements of system dynamics delay currently being defined or updated

RLDPAR - parameters for system dynamics delay

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Function Value -

RLMDB - output value of boxcar material delay

Programs Called

Verbs. None

Other. GETSET, PERMDS, SETSET

Input/Output Files Used

None.

RLMD1
R/SS/CS
1/74

RLMD1 (DLRIX, RATEIN)

General Description

Function which returns the updated output of the rate and delay represented by DLRIX, and updates the level currently in the delay. The delay function represented is Forrester's 1st order exponential for material (The DELAY1 Macro in Dynamo).

Arguments.

DLRIX - index representing a specific delay of a specific rate

RATEIN - current input rate

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

TSTEP - time step

NSTEP - number of time steps executed to date

NSTW - number of time steps to be used for accelerated warmup of flow rates subject to delay

RLDST - array of PDS dataset type names

RL - elements of system dynamics delay currently being defined or updated

RLDPAR - parameters for system dynamics delay

Program Inputs

From Calling Program.

Arguments - see above

In/RLSYS/:

RL - array of elements of the dataset of dataset type RLD and coordinate DLRIX

Program Outputs

To Permanent Attributes.

None.

(RLMD1-1)

To Calling Program.

Function Value -

RLMD1 - updated output rate for first order exponential material delay

Programs Called

Verbs. None

Other. GETSET, PERMDS, SETSET

Input/Output Files Used

None.

RLMD3 (DLRIX, RATEIN)

General Description

Function which updates output of delayed rate represented by DLRIX, using Forrester's 3rd order material delay (Macro DELAY3 in Dynamo).

Arguments.

DLRIX - index representing a specific delay of a specific rate

RATEIN - current input rate

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

TSTEP - time step

NSTEP - number of time steps executed to date

NSTW - number of time steps to be used for accelerated warmup of flow rates subject to delay

RLDST - array of PDS dataset type names

RL - elements of system dynamics delay currently being defined or updated

RLDPAR - parameters for system dynamics delay

Program Inputs

From Calling Program.

Arguments - see above

In/RLSYS/:

RL - array of elements of the dataset of dataset type RLD and coordinate DLRIX

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Function Value -

RLMD3 - updated output rate of 3rd order exponential material delay

Programs Called

Verbs. None

Other. GETSET, PERMDS, SETSET

Input/Output Files Used

None.

RLNIX (RLN)

General Description

Function which returns index of rate/level node name RLN in table RLNNAM, or zero if not present.

Arguments.

RLN - alpha node name for which index is being sought

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

RLNNAM - array of names of system dynamics nodes

NRLN - number of system dynamics nodes in current model

Program Inputs

From Calling Program.

Argument - see above

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

Function Value - RLNIX - index of RL node name RLN in table RLNNAM

Programs Called

Verbs. None

Other. None

Input/Output Files Used

None.

THIS PAGE LEFT INTENTIONALLY BLANK

RLRAT (ISPTR, NODE, DR, INOD)

General Description

Returns value of rate in rate/level stack pointed from ISPTR. The rate returned is that associated with node NODE. The rate is that whose node number is NODE if INOD = 0, the successor of that in the stack if INOD = +1, or the predecessor if INOD = -1. It is a demand stack if DR = D, a return flow stack if DR = R.

Arguments.

ISPTR - word from which pointer is unpacked, pointer to rate stack in DTABLE

NODE - the node number to search for in rate stack

DR - indicator for type of stack, D = demand, R = receive

INOD - 0, return rate from NODE entry

+1, return rate from successor to NODE entry

-1, return rate from predecessor to NODE entry

GASP Files Used

None.

Permanent Attributes Accessed

In /TBLCOM/:

DTABLE - an array for storing pairs of values, in this case rate stacks.

Program Inputs

From Calling Program.

Arguments as above

From IUNPAK.

N - element unpacked from left of rightmost

ILO - rightmost element unpacked

From IRSTK.

Function Value: index to NODE entry in rate stack in DTABLE

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

RLRAT - current value of the flow rate on the specified link

Programs Called

Verbs. None

Other. IRSTK, IUNPAK

Input/Output Files Used

None.

RLSASN (JRLST)

General Description

For a PDS dataset whose system dynamics statistics collection indicators have just been input into JRLST, assigns identifiers for each variable to be observed. The indicators for the Ith element of the dataset are in JRLST(I). They are a frequency of observation code, F, and a type of observation code, T, packed as (F*10) + T. JRLST(I) = 0 means do not collect statistics on Ith element of dataset.

When a variable is to be observed, identifiers and pointers are established in the next available position KRLST and KRLSTI in /RLSYS/ and in RLSNAM in /STNAM/. If the form of observation is COLCT, HISTO, or TACTN, identifiers are stored in CLTNAM, HNAM, or TCTNAM in /STNAM/ as well. The data stored in these arrays are described in Chapter III.

Arguments.

JRLST - system dynamics statistics collection indicators for the elements of a PDS dataset, in the form F*10 + T, where F is a frequency code 1, 2, ..., 9 indicating that interval RLSTIU(F) is pass between observations, and T is a code indicating type of observation - 1, print; 2, (default to print); 3, COLCT; 4, HISTO; 5, TACTN.

Examples

RLSASN is called by vRLPDS and vCHPDS after each dataset is read.

GASP Files Used

None.

Permanent Attributes Accessed

In /RLSYS/.

NXRLST - last position used in KRLST, KRLSTI, and RLSNAM

NRLST - dimension of KRLST, KRLSTI, and RLSNAM

In /SVCLCT/.

NXTCL - index of last set of COLCT counters assigned

NCOLCT - total number of COLCT counters in model

In /SVHSTO/.

NXTHST - index of last set of HISTO counters assigned

NHISTO - total number of HISTO counters in model

In /SVTCTN/.

NXTCTN - index of last position in TCTNAM assigned

NTCTN - total number of positions in TCTNAM

Program Inputs

From Calling Program.

Argument JRLST, collection indicators for elements of current PDS dataset.

In /DSNOW/ - IXEL1, index in DSPOOL of position immediately preceding 1st element of current dataset

DSETNC, name of current dataset type

From IXST1. Function value, IST, index of COLCT, HISTO, or TACTN name entry and, for COLCT and HISTO, counters, assigned to current variable.

From LRNIX (DSETNC). Index of current dataset type in table RLDST in /RLSYS/.

Program Outputs

To Permanent Attributes.

In /RLSYS/ -

NXRLST updated to reflect any new entries in KRLST, KRLST1, and RLSNAM

KRLST and KRLST1 with entries added for variables in current dataset

In /STNAM/ - entries added to RLSNAM, CLTNAM, HNAM, and TCTNAM as appropriate

In /SVCLCT/ - NXTCL updated to reflect new COLCT variables

In /SVHSTO/ - NXTHST updated to reflect new HISTO variables

In /SVTCTN/ - NXTCTN updated to reflect new TACTN variables

To Calling Program. None.

(RLSASN-2)

To IXST1 (next, max, nam, 0., I). For COLCT, HISTO, or TACTN collection types

Arguments -

next - last assigned name table position (and set of counters for COLCT or HISTO)

max - max available in model

nam - array of names

0. - PERMAT identifier (not applicable)

I - index of current variable in current dataset

In /INSTAT/ -

XIDST1 - LRNIX (DSETNC)*10000 + I*100 + 19, statistic variable identifier, in which I is its position in current dataset, LRNIX (DSENC) represents dataset type, and 19 indicates TMST (the 1) and system dynamics (the 9)

XIDRS1 - "resource" identifier, consisting of flow class number and, if appropriate, priority, as the primary and secondary parts, respectively

INOD1 - node number

To LRNIX (DSETNC). Argument DSETNC, name of current dataset type

Programs Called

Verbs. None.

Other. IPAK2, IXST1, LRNIX.

Input/Output Files Used

None.

(RLSASN-3)

THIS PAGE LEFT INTENTIONALLY BLANK

RLSRPT

General Description

Lists system dynamics variables subject to statistics collection and their frequency and collection type codes.

Arguments.

None.

GASP Files Used

None

Permanent Attributes Accessed

In/RLSYS/

RLDST - list of dataset type names

LRLST - index of last statistic to be observed

In/STNAM/

RLSNAM - array Rate/Level statistic names

In/DSTOR/

DSPOOL - index of data set descriptors

Program Inputs

From Calling Program.

None.

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

None.

To File NPRNT.

Report of all indentifiers of Rate/Level statistics and frequency of collection.

Programs Called

Verbs. None

Other. HASHI, IUNPAK, LIN, MSHIFT

Input/Output Files Used

NPRNT - Print File

RLSSET

General Description

Sorts system dynamics statistics array KRLST into order of descending frequency of observation (ascending frequency code), and strips off the frequency code. Before stripping off the frequency code, array LRLST is set. LRLST(I) is the last position for frequency code I in the sorted KRLST. Prints report of collection intervals and number of variables for each. Also prints system dynamics statistics identifiers via RLSRPT. RLSSET schedules the first RLSTAT collection event when first called. Control is returned to the calling routine. When RLSSET is called subsequent to the initial call, frequency codes are first pub back into KRLST so that the sorting can proceed properly.

Arguments

None.

Examples

RLSSET is called by vRLPDS and vCHRLD after all statistics collection indicators have been read and processed by RLSASN.

GASP Files Used

Time File.

Permanent Attributes Accessed

In /RLSYS/ - KRLST, array of frequency and type codes for system dynamics statistics
NXRLST, number of active entries in KRLST
LRLST(1), - 1 if this is first RLSSET, ≥ 0 if not
LRLST - if not first call, previous frequency pattern, now to be revised

Program Inputs

From Calling Program. None.

From SORTI (KRLST, NXRLST). KRLST in ascending order.

From FILEM (1, ATRIB). None.

From RLSRPT. None.

AD-A040 804 MODELS OF THE US ARMY WORLDWIDE LOGISTIC SYSTEM (MAWLOGS) VOLUME III MODU..(U) BDM CORP VIENNA VA 4/4

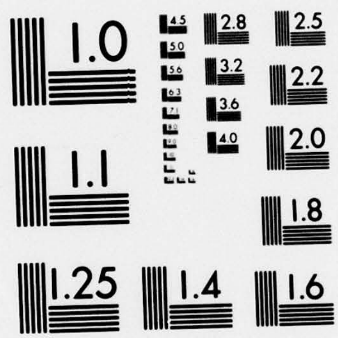
FEB 77 BDM/W-76-211-TR-VOL-3-PT-6 DAAG39-76-C-0134

UNCLASSIFIED F/G 15/5 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Program Outputs

To Permanent Attributes. In /RLSYS/ - revised KRLST and LRLST

To SORTI (KRLST, NXRLST). Arguments KRLST, array to be sorted, and NXRLST, number of entries.

To FILEM (1, ATRIB). Arguments 1, time file, and ATRIB

ATRIB(1) - TNOW + 1.01*RLSTIU(1), time for first RLSTAT event (.01 is to insure it follows rate and level updating)

ATRIB(2) - -7, GASP event number for RLSTAT

ATRIB(3) - LRLST(1), index in KRLST of last variable to be observed when this event occurs

ATRIB(4) - 1, frequency code for variables with lowest frequency of observation to be covered by this event

To RLSRPT. None.

To Calling Program. None.

Programs Called

Verbs. None.

Other. FILEM, PGHDR, RLSRPT, SORTI.

Input/Output Files Used

NPRNT - print file.

(RLSSET-2)

RLSTAT

General Description

Supervisor of system dynamics statistics collection in a MAWLOGS model. At each execution, all system dynamics variables subject to observation at specified frequencies are observed. Their current values are either printed or saved as time weighted values in COLCT, HISTO, or TACTN form. The variables to be observed are identified in arrays KRLST and KRLST1 in /RLSYS/. Their frequency of observation is specified by arrays RLSTIU and LRLST in /RLSYS/. Execution of RLSTAT is scheduled via the time file as GASP event -7. The first execution is scheduled by vRLPDS. Subsequent executions are scheduled by RLSTAT itself. The variables to be observed and the frequency and form for each are specified in vRLPDS and vCHPDS inputs. Currently, only variables stored in PDS datasets can be observed. At the conclusion of an RLSTAT execution, control is returned to subroutine EVNTS.

Arguments

None.

Statistics Collected

All system dynamics statistics in a model.

GASP Files Used

Time file.

Permanent Attributes Accessed

In/RLSYS/:

KRLST(I) - array specifying COLTYP, collection form; and IXDSPL, location of variable in DSPOOL; also MRLST, index of sister entry in KRLST 1; I=1, 2, ..., LAST, where LAST was obtained as ATRIB(3) of GASP event -7.

KRLST1 - array specifying IX, index in COLCT, HISTO, or TACTN identifiers for all system dynamics statistical variables

LRLST(I) - index in KRLST Of last variable to be observed with frequency corresponding to RLSTIU(I), I = 1,9

RLSTIU(I) - interval between observations for variables subject to the Ith collection frequency, I = 1, 2, ..., 9; RLSTIU(I) = (2**(I-1))*RLSTIU(1) I = 2, 3, ..., 9. RLSTIU(1) is input via vRLPDS.

In DSPOOL:

DSPOOL - values of variables to be observed

Program Inputs

From Calling Program.

In/VRBGSP/, attributes of current GASP event -7, in ATRIB:

ATRIB(1) - TNOW

ATRIB(2) - -7

ATRIB(3) - LAST, index in KRLST of last variable to be observed in current execution of RLSTAT

ATRIB(4) - LNOW, index in LRLST of entry corresponding to LAST

From FILEM (1, ATRIB).

None.

From PRRLST.

None.

From COLCT.

None.

From HISTO.

None.

From TACTN.

None.

Program Outputs

To Permanent Attributes.

None.

To Calling Program.

None.

To FILEM (1, ATRIB).

Argument ATRIB, attributes of next RLSTAT event

ATRIB(1) - TNOW + RLSTIU(1)

ATRIB(2) - -7, event code

ATRIB(3) - index in KRLST of last index to be observed next time

ATRIB(4) - index in LRLST of entry corresponding to ATRIB(3)

To PRRLST.

In/RLSYS/ - IRLST, index in KRLST of variable to be printed; MRLST, index of corresponding entry in KRLST1; and IXDSPL, index in DSPOOL of current value; also PRRLSW, page header control switch.

To COLCT, HISTO and TACTN.

In/INSTAT/ - IX, index of counters to update; X1, current value of variable; X2, time since last observation.

Programs Called

Verbs. None

Other. COLCT, FILEM, PRRLST, TACTN

Input/Output Files Used

NPRNT - print file, via PRRLST

NTACTN - transaction file, via TACTN

SAVLK (IZLDSC)

General Description

This routine saves the flow network data structure from the referenced form to the extension of the intital definition form in the pushdown stacks so that changes may be made.

If removes LKDST entries for LDSC(RLNODE) (whose pointer word is argument IZLDSC) from DTABLE to IZHOLD, and the associated pointers to RLC from LCIX to IZHOLD.

Returns as IZLDSC the negative location in IZHOLD of the stack of LKDST entries. LCIX and DTABLE space is freed after this operation.

Arguments.

IZLDSC - pointer word to LKDST entries in DTABLE for dataset LDSC (rlnode).

GASP Files Used

None.

Permanent Attributes Accessed

In/TBLCOM/:

DTABLE - a table of linked-rate dataset type names with pointers into LCIX array.

LCIX - pointers to RLC coordinate array

Program Inputs

From Calling Program.

Argument -

IZLDSC - pointer word to DTABLE from LDSC (node) dataset (contains index of first entry in DTABLE and number of dataset type names).

From ZPLAEN.

In/ZSTACK/:

IZREF - pointer to top of stack being created in IZHOLD. Two types of stacks are created, one for pointers from LCIX array and one for dataset name from DTABLE table.

Program Outputs

To ZPLAEN (first call).

In/ZSTACK/: entries for LCIX content stacks

IZKC - IPAK2 (NREFS, NCDS) where NREFS is the number of references to the node and NCDS is the number of dataset coordinates

IZK - position of the coordinates in array RLC

To ZPLAEN (second call).

In/ZSTACK/: entries for DTABLE dataset names

IZKC - name of the dataset from DTABLE (1,·)

IZK - IPAK2 (NXLC, IZREF) where NXLC is the number of entries in the stack and IZREF is the pointer to the top entry in the LCIX stack.

To RELCIX.

Arguments -

NXLC - number of entries to release from LCIX

IXLC1 - first entry in LCIX stack

To RLDTB.

Arguments -

NDSTP - number of entries in DTABLE table to release

IDSTP - first entry in table in DTABLE

To Calling Program.

Arguments -

IZLDSC - negative pointer to IZHOLD pushdown stack of LKDST entries

(SAVLK-2)

In/LC/:

LRLCHG - signifies to LRLOOP that LCIX list has been transferred to IZHOLD stack

LRLPTR - pointer to stack in IZHOLD containing LCIX list

Programs Called

Verbs. None

Other. IPAK2, IUNPK2, ZPLAEN, RELCIX, RLDTB

Input/Output Files Used

None.

TRSTAK (IPTRW)

General Description

Transfers from DTABLE to IZHOLD the rate/level rate stack pointed to from IPTRW. TRSTAK is set to the negative index (integer) in IZHOLD of a pointer entry. The pointer entry points to the transferred rate stack with a negative pointer in IZKC. IZK of this entry is reserved for a negative pointer to a new rate stack. The DTABLE space occupied by the old (current) rate stack is released.

Arguments.

IPTRW - pointer to indicate the RL rate stack to be transferred from DTABLE to IZHOLD

GASP Files Used

None.

Permanent Attributes Accessed

In /DSNOW/:

NE - number of elements in dataset of current interest

IXEL1 - index in DSPOOL of first element in dataset of current interest

In /RLSYS/:

RLDST - array containing list of PDS dataset type names

In /TBLCOM/:

DTABLE - an array for storing pairs of values

PDS Dataset RATATT:

Elements saved in stack entry and dataset released

Program Inputs

From Calling Program.

Argument - see above

From RLDTB. None.

Program Outputs

To Permanent Attributes. None.

(TRSTAK-1)

To RLDTB.

Arguments

NR - number of elements to be released

IR1 - first element to be released

To Calling Program.

TRSTAK - negative index in IZHOLD of a pointer entry. The pointer entry points to the transferred rate stack with a negative pointer in IZKC.

Programs Called

Verbs. None.

Other. GETSET, IPAK2, IUNPK2, PERMDS, RELSET, RLDTB, ZPLAEN.

Input/Output Files Used

None.

UDDR (RATEN, INOD, ISTKP, BOTSW)

General Description

Updates demand rate in RL rate stack entry at RIXDT, location in DTABLE of an entry in the demand rate stack pointed to by ISTKP. The particular entry is determined from INOD and BOTSW in accordance with the following rules -

INOD.GT.0 - The successor entry to that associated with INOD

INOD.LT.0 - The entry associated with -INOD

INOD = 0 - The entry associated with KRLNOD, the current RLNOD

BOTSW.NE.0 - The first (bottom) entry in the stack if the entry indicated by the above conventions is not found

BOTSW = 0 - If entry is not found in accordance with INOD conventions, control returns to the calling routine, i.e., the bottom entry option is not taken

The rate used is RATEN and RATINC where RATINC is the rate increment from RATATT.1

Arguments.

RATEN - new rate to place in the stack entry

INOD - node number associated with an entry in the stack as described above

ISTKP - pointer to a rate stack in DTABLE

BOTSW - bottom switch, as described above

Examples

UDDR is called by verbs such as UDDMD to update a particular demand rate stack entry.

GASP Files Used

None.

Permanent Attributes Accessed

In/RLSYS/:

KRLNOD - current RL node number

In/TBLCOM/:

DTABLE - array for storing pairs of values

PDS Dataset RATATT (RIXDT).1 - value to increment the rate stack entry at RIXDT by.

Program Inputs

From Calling Program.

Arguments - as above

Program Outputs

To Permanent Attributes.

In COMMON/SUPC/:

RIXDT - index to the rate stack entry in DTABLE which was updated

To Calling Program.

None.

Programs Called

Verbs. None

Other. IRSTK, IUNPAK, PERMDS, RLDLY, SETEL, VALEL

Input/Output Files Used

None.

UDDRA1 (RATEN, INOD, ISTKP, BOTSW)

General Description

This routine is used to increment a demand rate entry by adding value RATEN to the PDS dataset RATATT (RIXDT).1 where RIXDT is the location in DTABLE of an entry in the demand rate stack pointed to by ISTKP. RIXDT is determined from the arguments INOD and BOTSW as described below. The cumulative value in RATATT.1 is added to the demand rate when the next UDDR is performed. This allows multiple updates to a demand stack during a single time step, a process that is not possible with UDDR since the stack entry is overwritten each time. If a RATATT dataset does not exist for the stack entry, a new one is created.

Arguments.

RATEN - rate value to be added to demand rate

ISTKP - pointer to rate stack in DTABLE to search

INOD - the particular entry is determined from INOD and BOTSW in accordance with the following rules -

INOD.GT.0 - the successor entry to that associated with INOD

INOD.LT.0 - the entry associated with -INOD

INOD = 0 - the entry associated with KRLNOD, the current RLNOD

BOTSW - bottom switch

BOTSW.NE.0 - the first (bottom) entry in the stack if the entry indicated by the above conventions is not found

BOTSW = 0 - if entry is not found in accordance with INOD conventions, control returns to the calling routine, i.e., the bottom entry option is not taken

GASP Files Used. None.

Permanent Attributes Accessed

PDS Datasets:

RATATT (RIXDT).1 - rate increment, where RIXDT is the index of the rate stack entry in DTABLE specified by INOD and BOTSW

(UDDRA1-1)

Program Inputs

From Calling Program.

In /RLSYS/:

KRLNOD - current rate/level node number

Arguments as specified above

From IRSTK.

Function Value - index to the demand rate in DTABLE

Program Outputs

To Permanent Attributes.

PDS Datasets:

RATATT (RIXDT).1 - cumulative demand rate increment. This dataset will be created if it does not exist

To IRSTK.

Arguments -

NOD - node number associated with stack entry

ISTKP - pointer to rate stack in DTABLE

To Calling Program. None.

Programs Called

Verbs. None.

Other. IRSTK, PERMDS.

Input/Output Files Used. None.

(UDDRA1-2)

UDRR (RATEN, INOD, ISTKP, TOPSW)

General Description

Updates return rate for an entry in a rate/level rate stack from RATEN and RATATT.1. The return flow rate in question is one in the stack pointed to from ISTKP. RATEN is the rate to be used for input to the rate stack. The particular stack entry to be updated is determined by INOD and TOPSW as follows -

INOD.GT.0 - The successor to the entry associated with INOD is updated

INOD.LT.0 - The entry associated with -INOD is updated

INOD = 0 - The entry associated with KRLNOD is updated

TOPSW controls the case where the desired entry as expressed by INOD cannot be found

TOPSW=0. - No further action (no entry updated)

TOPSW=1. - Top entry - i.e., the entry furthest upstream in the return flow - is updated

Arguments.

RATEN - rate value to update return rate stack entry

INOD - node associated with a rate stack entry as described above

ISTKP - pointer word to rate stack in DTABLE

TOPSW - top switch as described above

Examples

UDRR is called by UDRAT to update the return rate entry from a node, including any rate increment contained in dataset RATATT.1.

Call UDRR (IFSRI(M) + IABORI(M), KRLNOD, IRPTR(IXRP),0.)

GASP Files Used

None.

Permanent Attributes Accessed

In/TBLCOM/:

DTABLE - rate stack to be updated

PDS Dataset RATATT(RIXDT).1 - rate increment where RIXDT is the index of the rate stack entry in DTABLE

Program Inputs

From Calling Program.

Arguments - as specified above

Program Outputs

To Permanent Attributes.

PDS Dataset RATATT(RIXDT).1 is cleared to zero.

To IRSTK.

Arguments -

First - node number associated with stack entry

Second - pointer to stack in DTABLE

To RLDLY.

Arguments -

First - index of rate/level delay to apply

Second - value to add to delay, RATEN + RATINC

To Calling Program.

None.

Programs Called

Verbs. None

Other. IRSTK, IUNPAK, PERMDS, RLDLY

Input/Output Files Used

None.

(UDRR-2)

UDRRA1 (RATEN, INOD, ISTKP, TOPSW)

General Description

This routine is utilized in the same manner as UDRR to add a value to a return rate entry. However, this routine adds the value to the RATATT dataset associated with the rate stack entry. The value in RATATT.1 is then added to the rate as an increment when the next UDRR is performed. This allows rate adjustments to be made by more than one module during a time step, since UDRR obliterates the existing value in the stack entry (presumably for the previous time step). If a RATATT dataset does not exist, one is created.

Arguments.

RATEN - the rate to be used for input to the rate

ISTKP - pointer to the return rate stack to search

INOD - the particular stack entry to be updated is determined by INOD and TOPSW as follows -

INOD.GT.0 - the successor to the entry associated with INOD is updated

INOD.LT.0 - the entry associated with -INOD is updated

INOD = 0 - the entry associated with KRLNOD is updated

TOPSW - controls the case where the desired entry as expressed in INOD cannot be found

TOPSW = 0. - no further action (no entry updated)

TOPSW = 1. - top entry - i.e., the entry furthest upstream in the return flow - is updated

GASP Files Used. None.

Permanent Attributes Accessed

PDS Datasets:

RATATT (RIXDT).1 - rate increment, where RIXDT is the index to the rate stack entry in DTABLE to be incremented

Program Inputs

From Calling Program.

Arguments as specified above

In /RLSYS/:

KRLNOD - rate/level node number

From IRSTK.

Function Value - index to the return rate in DTABLE

Program Outputs

To Permanent Attributes.

PDS Datasets:

RATATT (RIXDT).1 - cumulative return rate increment. This dataset will be created if it does not exist

To IRSTK.

Arguments -

NOD - node number associated with stack entry

ISTKP - pointer to rate stack in DTABLE

To Calling Program. None.

Programs Called

Verbs. None.

Other. IRSTK, IUNPARK, PERMDS.

Input/Output Files Used. None.

(UDRRA1-2)

XLINE (IPARAM, X)

General Description

Interpolates in the table identified with PARAM index IPARAM. The table is thought of as a function, not necessarily a distribution function, represented with linear segments.

X is the abscissa value for which the ordinate is to be determined. Entry PARAM(IPARAM.*) contains

1 - Blank

2 - number of points in table

3 - location in DTABLE array

4 - 0, indicating a continuous function

Arguments.

IPARAM - index of PARAM array entry which identifies table of values to be interpolated

X - abscissa value for which ordinate is to be determined

GASP Files Used

None.

Permanent Attributes Accessed

In /TBLCOM/:

DTABLE - array for storing pairs of values

In /VRBGSP/:

PARAM(I,J) - array containing parameters for the tabular function, identified by I, found in DTABLE. Entries for J are described above.

Program Inputs

From Calling Program.

Arguments - see above

Program Outputs

To Permanent Attributes. None.

To Calling Program.

Function Value - XLINE - ordinate determined from interpolation of table for abscissa value X

Programs Called

Verbs. None.

Other. None.

Input/Output Files Used

None.

```
                    ** MAWLOGS MODEL DSS4           **                    ANALYST.,   BRM
                                                                          DATE.     10/ 18/ 1975
                          SUMMARY OUTPUT                                  RUN NAME.(VII/W/Z) SLICE
                                                                          TNOW      160.000
              FLOWS IN RATE/LEVEL LINKS OF TYPE FCRAT  AT TIME   160.000

                                             RLNODE  RLNODE                  DELAY   DELAY   DELAY      DELAY
                                             NUMBER  NAME   RATE             NUMBER  TYPE    PARAMETER  CONTENT

*** OWNER NODE IS NUMBER   46 NAMED 67IN         FLOW CLASS IS     902.000                                    ***
    RATE STACK POINTER FCRAT . 3.0/RPTR =0000301471
                                               46  67IN    8.30790E+03       0                 0.000   0.
                                               64  9GS     0.                0                 0.000   0.
                                               80  CONUS   7.25177E+03       0                 0.000   0.
                                                                                                       0.            SUM
    RATE STACK POINTER FCRAT . 4.0/RPTR =0000100755
                                               79  WRSL    0.                0                 0.000   0.
                                                                                                       0.            SUM
    RATE STACK POINTER FCRAT . 5.0/RPTR =0000200756
                                               46  67IN    2.52615E+03     615   RLMD3        46.500   2.47034E+05
                                               81  CCPE    4.89322E+03     614   RLMD1         7.000   3.42525E+04
                                                                                                       2.81287E+05   SUM
    RATE STACK POINTER FCRAT . 7.0/RPTR =0000100760
                                               46  67IN    0.              616   RLMD3        21.000   0.
                                                                                                       0.            SUM

*** OWNER NODE IS NUMBER   46 NAMED 67IN         FLOW CLASS IS     901.000                                    ***
    RATE STACK POINTER FCRAT . 3.0/RPTR =0000301474
                                               46  67IN    1.89210E+03       0                 0.000   0.
                                               64  9GS     0.                0                 0.000   0.
                                               80  CONUS   1.04522E+02       0                 0.000   0.
                                                                                                       0.            SUM
    RATE STACK POINTER FCRAT . 4.0/RPTR =0000100750
                                               79  WRSL    0.                0                 0.000   0.
                                                                                                       0.            SUM
    RATE STACK POINTER FCRAT . 5.0/RPTR =0000200751
                                               46  67IN    2.08567E-02     611   RLMD3        16.500   2.32001E+03
                                               81  CCPE    2.15992E+02     610   RLMD1         5.300   1.13999E+03
                                                                                                       3.45999E+03   SUM
    RATE STACK POINTER FCRAT . 6.0/RPTR =0000100753
                                               46  67IN    1.28770E+03     612   RLMD1         7.000   9.01392E+03
                                                                                                       9.01392E+03   SUM
    RATE STACK POINTER FCRAT . 7.0/RPTR =0000100754
                                               46  67IN    0.              613   RLMD3        11.000   2.54084E-02
                                                                                                       2.54084E-02   SUM

*** OWNER NODE IS NUMBER   46 NAMED 67IN         FLOW CLASS IS     402.000                                    ***
    RATE STACK POINTER FCRAT . 3.0/RPTR =0000301477
                                               46  67IN    2.87675E+04       0                 0.000   0.
                                               64  9GS     0.                0                 0.000   0.
                                               80  CONUS   3.20431E+03       0                 0.000   0.
                                                                                                       0.            SUM
    RATE STACK POINTER FCRAT . 4.0/RPTR =0000100744
                                               79  WRSL    0.                0                 0.000   0.
                                                                                                       0.            SUM
    RATE STACK POINTER FCRAT . 5.0/RPTR =0000200745
                                               46  67IN    1.13310E+04     608   RLMD3        46.500   1.19182E+06
                                               81  CCPE    1.48515E+04     607   RLMD1         7.000   1.03995E+05
                                                                                                       1.29582E+06   SUM
```

Figure III-1. Flows in Rate/Level Links Report